TimeSiam: A Pre-Training Framework for Siamese Time-Series Modeling

Jiaxiang Dong [*1] **Haixu Wu** [*1] **Yuxuan Wang** [1] **Yunzhong Qiu** [1] **Li Zhang** [1] **Jianmin Wang** [1] **Mingsheng Long** [1]

Abstract

Time series pre-training has recently garnered wide attention for its potential to reduce labeling expenses and benefit various downstream tasks. Prior methods are mainly based on pre-training techniques well-acknowledged in vision or language, such as masked modeling and contrastive learning. However, randomly masking time series or calculating series-wise similarity will distort or neglect inherent temporal correlations crucial in time series data. To emphasize temporal correlation modeling, this paper proposes TimeSiam as a simple but effective self-supervised pre-training framework for Time series based on Siamese networks. Concretely, TimeSiam pre-trains Siamese encoders to capture intrinsic temporal correlations between randomly sampled past and current subseries. With a simple data augmentation method (e.g. masking), TimeSiam can benefit from diverse augmented subseries and learn internal time-dependent representations through a past-to-current reconstruction. Moreover, learnable lineage embeddings are also introduced to distinguish temporal distance between sampled series and further foster the learning of diverse temporal correlations. TimeSiam consistently outperforms extensive advanced pre-training baselines, demonstrating superior forecasting and classification capabilities across 13 standard benchmarks in both intra- and cross-domain scenarios. Code is available at https://github.com/thuml/TimeSiam.

1. Introduction

Time series, a critical form of real-world data, finds wide applications in various domains, including energy, traffic, economics, weather, medicine, etc (Wu et al., 2021; Zhang et al., 2022; Wu et al., 2023b). In the real world, an enormous volume of time series data is incrementally collected through the Internet of Things (IoT) from industrial sensors and wearable devices. To utilize these large amounts of data, time series self-supervised pre-training has recently gained significant attention, which can extract valuable knowledge from unlabeled data and further boost the performance of various downstream tasks (Dong et al., 2023). This paper focuses on this promising area and proposes a novel and practical self-supervised pre-training method for time series.

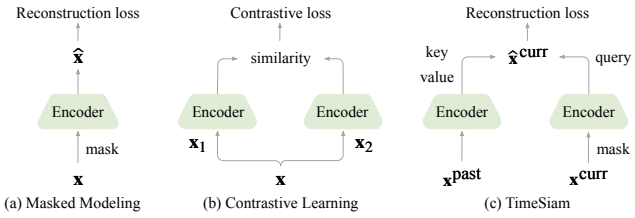

Figure 1. Comparison on time series pre-training frameworks. (a) Masked modeling: reconstruct the masked series. (b) Contrastive learning: Repulse different series (negative pairs) while attracting two augmentations from the same series (positive pairs). (c) TimeSiam: reconstruct masked current series $\mathbf{x}^{\text{curr}}$ from randomly sampled past observation $\mathbf{x}^{\text{past}}$.

Previous pre-training methods can be roughly categorized into the following two paradigms. As presented in Figure 1, the first one, named masked modeling, enables representation learning by optimizing the model to reconstruct the masked part using the visible context, which has been commonly used in natural language processing (Devlin et al., 2018; Raffel et al., 2020) and computer vision (He et al., 2022; Xie et al., 2022; Li et al., 2023a). However, Dong et al. (2023) found that randomly masking a portion of time points will seriously distort vital temporal correlations of time series, making the reconstruction task too difficult to guide representation learning. The other paradigm, contrastive learning, excels in instance-level representation learning, which optimizes the model to identify positive samples from negative ones (Tang et al., 2020). A significant criticism of these contrastive approaches is their reliance on careful augmentations selection to learn useful invariances (Xiao et al., 2021), which is even harder in time series due to the scarcity of widely accepted and effective data augmentation methods (Wen et al., 2020). Also, the instance-level modeling design may fail in capturing fine-grained temporal

[*]Equal contribution [1]School of Software, BNRist, Tsinghua University. Jiaxiang Dong <djx20@mails.tsinghua.edu.cn>. Haixu Wu <wuhx23@mails.tsinghua.edu.cn>. Correspondence to: Mingsheng Long <mingsheng@tsinghua.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

variations, limiting their practicality to downstream tasks.

We note a crucial distinction of time series from images or languages, as each time step consists of only a finite number of scalar values. This implies that the most vital information in time series is preserved in the temporal correlations, highlighting the importance of temporal modeling (Nie et al., 2023; Wu et al., 2023a). Therefore, *the critical point of time series pre-training is optimizing encoders to accurately capture temporal correlations*, which has not been adequately addressed in previous masking or contrastive methods.

To address the insufficiency in temporal modeling, we present TimeSiam, a simple yet effective self-supervised pre-training framework. Unlike prior, as shown in Figure 1, TimeSiam proposes to sample pairs of subseries across different timestamps from the same time series, termed "*Siamese subseries*". Then, it leverages Siamese networks as encoders to capture correlations between temporally distanced subseries. With simple data augmentation such as masking, TimeSiam further improves the diversity and distinctiveness of Siamese subseries, which natively derives a past-to-current reconstruction task, thereby enforcing the encoder to learn temporally related information and capture correlations among past and current series. Besides, to cover different distanced Siamese subseries, we propose learnable lineage embeddings to enhance the encoder capacity for learning diverse time-dependent representations. Eventually, a decoder that integrates cross-attention and self-attention mechanisms is applied to ensure a precise reconstruction of the (masked) Siamese subseries.

Importantly, TimeSiam is not constrained by proximity information in the time series. Instead, benefiting from our Siamese subseries sampling procedure, it can effectively model the correlation among distanced subseries, which empowers the model with a more thorough understanding of the whole time series. With the above designs, TimeSiam remains simple but achieves consistent state-of-the-art against prior time series pre-training methods across various downstream tasks, including time series forecasting and classification, covering both in- and cross-domain settings. Overall, our contributions are summarized as follows:

- In the spirit of learning temporal correlations, we propose TimeSiam, a simple but effective pre-training framework that leverages Siamese networks to capture correlations among temporally distanced subseries.
- With Siamese encoders to reconstruct current masked subseries based on past observation and lineage embeddings to capture subseries disparity, TimeSiam can learn diverse time-dependent representations.
- TimeSiam achieves consistent state-of-the-art fine-tuning performance across thirteen standard benchmarks, excelling in various time series analysis tasks.

2. Related Work

2.1. Time Series Self-supervised Pre-training

Self-supervised pre-training has demonstrated its ability to learn valuable and generalizable representations from large-scale unlabeled datasets in various domains, such as natural language processing (NLP) (Devlin et al., 2018; Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020; Gao et al., 2020) and computer vision (CV) (He et al., 2020; Liu et al., 2021; Xie et al., 2022; He et al., 2022), which can significantly reduce labeling expenses and benefit diverse downstream tasks. Recently, self-supervised pre-training has empowered many breakthroughs in time series analysis by introducing well-established techniques into time series, such as masked modeling and contrastive learning.

Masked Modeling As a fundamental technique in self-supervised pre-training methods, masked modeling enables deep models to learn essential representations by reconstructing masked parts from the visible context. Drawing inspiration from notable advances in NLP and CV, extensive time series pre-training approaches focus on time series masked modeling, which helps the model learn effective time series representations to facilitate various downstream analysis tasks. For instance, TST (Zerveas et al., 2021) and Ti-MAE (Li et al., 2023b) propose to randomly mask segments and points in time series and pre-train the model with the reconstruction task. PatchTST (Nie et al., 2023) divides the temporal dimension into multiple patches and treats time series as independent variates. Additionally, it incorporates a non-overlapping patch-level masked self-supervised strategy for temporal representation learning. HiMTM (Zhao et al., 2024) proposes a novel hierarchical masked time series pre-training method to capture the multi-scale nature of time series. Additionally, SimMTM (Dong et al., 2023) introduces a multi-masking modeling paradigm, which reconstructs original time series through the weighted aggregation of multiple masked time series, thus being able to learn both points-wise and series-wise temporal representations. Despite these advances, all of these approaches solely focus on the modeling of one individual time series, disregarding the intrinsic temporal correlations and dynamical variations of the whole time series. In contrast, TimeSiam proposes to reconstruct the current sub-series based on past observations, which can naturally integrate the time-dependent information during reconstruction pre-training.

Contrastive Learning Unlike masked modeling, this approach enables model pre-training by optimizing the similarity among instance-level representations. It leverages different data augmentations to construct positive and negative pairs from data, where positive pairs are optimized to be close to each other and negative pairs are encouraged to be distant from each other during pre-training (Tang

et al., 2020; He et al., 2020; Chen & He, 2021; Gao et al., 2021). Current time series contrastive learning methods are mainly based on diverse data augmentations tailored to the domain-specific characteristics of time series. CPC (Oord et al., 2018) introduced contrastive predictive coding, which uses model-predicted timesteps as positive samples and randomly-sampled timesteps as negative samples to obtain advantageous time series representations for downstream tasks. Franceschi et al. (2019) combined a causal dilated convolutions-based encoder with a novel triplet loss that employs time-based negative sampling. TNC (Tonekaboni et al., 2021) learns the representations by ensuring that signals from within a neighborhood are distinguishable from the distribution of non-neighboring signals in the latent space using a debiased contrastive loss. TS2Vec (Yue et al., 2022) divides time series into patches, defining contrastive tasks at both the individual instance and patch levels. Mixing-up (Wickstrøm et al., 2022) exploits a data augmentation scheme in which new samples are generated by mixing two data samples. LaST (Wang et al., 2022) aims to separate seasonal and trend components in time series data within the latent space. Additionally, CoST (Woo et al., 2022) utilizes contrastive losses in both time and frequency domains to learn distinct seasonal and trend representations. Furthermore, TF-C (Zhang et al., 2022) introduces a novel time-frequency consistency architecture, optimizing for proximity between time-based and frequency-based representations of the same data sample. However, existing contrastive learning methods for time series heavily rely on intricate data augmentation techniques to generate diverse views of the original data for self-supervision. Also, the instance-level representation learning may fall short in downstream low-level tasks. In TimeSiam, we utilize the native temporally distanced subseries to build reconstruction tasks, thereby freeing from complex augmentation techniques and also considering the low-level representation.

2.2. Siamese Networks

Siamese networks (Bromley et al., 1993) are particular neural network architectures with shared model parameters. This design makes Siamese networks well-suited for comparing and distinguishing two input samples based on a single neural network. They have been widely used in contrastive learning to model the relationship between paired samples (Chen & He, 2021). The combination of Siamese networks and contrastive learning has been widely used in many applications, particularly in tasks requiring instance-level representations (Chen et al., 2020; He et al., 2020; Wang et al., 2023). However, in the field of time series pre-training, this combination generally focuses on recognizing subtle differences between various augmented views of the series itself, overlooking the essence of time series, that is temporal correlation modeling. In this paper, we explore using shared-weight Siamese autoencoders to establish correlations between past and current subseries. This methodology enables a more efficient understanding of temporal relations in time series and enforces the model to learn time-dependent representations.

3. TimeSiam

To enhance the time-dependent representation learning, TimeSiam is designed to capture correlations between temporally distant subseries based on Siamese networks. This framework can natively derive a past-to-current reconstruction task with simple masked augmentation. In addition, learnable lineage embeddings are incorporated to dynamically capture the disparity among different distanced subseries pairs, which can enhance the model's capacity to cover different temporal correlations. Hereafter, we will detail the pre-training and fine-tuning stages in TimeSiam.

3.1. Pre-training

TimeSiam pre-training involves the following two modules: Siamese subseries sampling and Siamese modeling.

Siamese Subseries Sampling Typically, previous time series pre-training approaches focus solely on modeling the individual series itself, neglecting the inherent correlations among temporally related time series. This deficiency in the pre-training phase will lead to insufficient extraction of generalizable time-dependent representations. In contrast, our TimeSiam is designed to focus on modeling temporal correlations of subseries across different timestamps, capturing the intrinsic time-correlated information of time series.

As shown in Figure 2, we construct Siamese subseries pairs by randomly sampling a past sample $\mathbf{x}^{\text{past}}$ preceding the current sample $\mathbf{x}^{\text{curr}}$ in the same time series. Each sample in a Siamese pair, termed "*Siamese subseries*" each other, contains T timestamps and C observed variables. Notably, one $\mathbf{x}^{\text{curr}}$ can correspond to multiple $\mathbf{x}^{\text{past}}$ subseries due to the random sampling process. We focus on constructing correlations and capturing temporal variations between these Siamese subseries, which benefits intrinsically time-dependent representation learning during pre-training. The relative distance between the past and current subseries, denoted as d, is crucial in representing the correlation and disparities between Siamese subseries. Furthermore, we adopt a simple masking augmentation to generate augmented current subseries $\widetilde{\mathbf{x}}^{\text{curr}}$ that further improves the diversity and the disparity of Siamese subseries pairs, ensuring a more robust and sufficient pre-training phase. The above process can be formalized as follows:

$$(\mathbf{x}^{\text{past}}, \widetilde{\mathbf{x}}^{\text{curr}}) = \text{Mask-Augment}\left((\mathbf{x}^{\text{past}}, \mathbf{x}^{\text{curr}})\right). \quad (1)$$

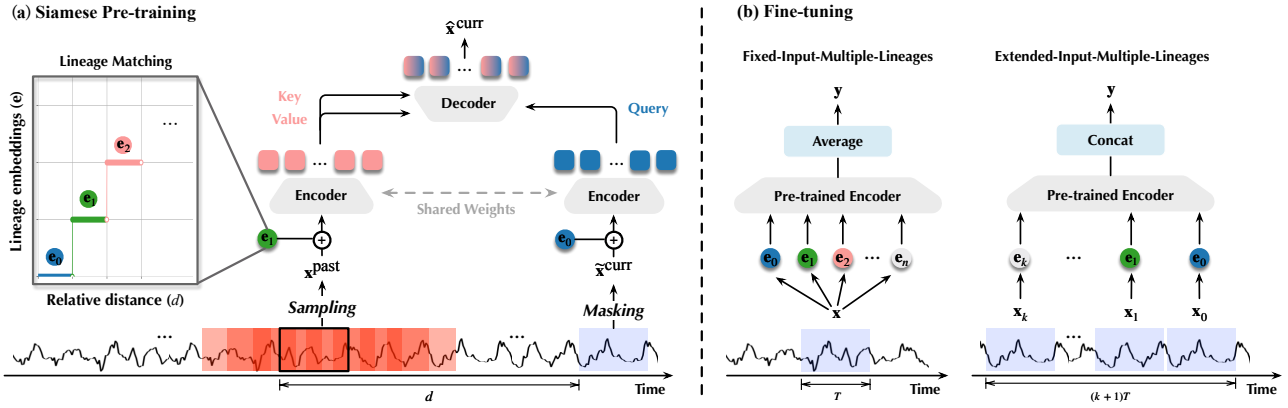

Figure 2. The overall design of TimeSiam, which establishes correlations between subseries randomly sampled from different timestamps using Siamese encoders. It integrates learnable lineage embeddings to enhance the capacity for temporal-related representation learning.

Siamese Modeling After constructing mask-augmented Siamese pairs, as shown in Figure 2, we further integrate learnable lineage embeddings during pre-training to effectively capture the disparity among different Siamese pairs. This design can enhance the model's capacity to extract diverse temporal-related representations. Given N learnable lineage embeddings $\{\mathbf{e}_i^{\text{lineage}}\}_{i=1}^{N}, \mathbf{e}_i^{\text{lineage}} \in \mathbb{R}^{1\times D}$ and D represents the dimension of lineage embeddings. For the past sample $\mathbf{x}^{\text{past}}$, we apply the $\text{LineageMatching}(\cdot)$ function to dynamically match a certain lineage embedding based on its temporal distance d to the current series. As for the current sample $\widetilde{\mathbf{x}}^{\text{curr}}$, we use a special lineage embedding to represent a degeneration situation as $d = 0$:

$$\begin{aligned} \mathbf{e}_i^{\text{lineage}} &= \text{LineageMatching}(d) \\ \mathbf{z}^{\text{past}} &= \text{Embed}(\mathbf{x}^{\text{past}}) \oplus \mathbf{e}_i^{\text{lineage}} \\ \widetilde{\mathbf{z}}^{\text{curr}} &= \text{Embed}(\widetilde{\mathbf{x}}^{\text{curr}}) \oplus \mathbf{e}_0^{\text{lineage}}, \end{aligned} \tag{2}$$

where $\mathbf{e}_0^{\text{lineage}} \in \mathbb{R}^{1\times D}$ is the specific embedding for current subseries and $\mathbf{z}^{\text{past}}, \widetilde{\mathbf{z}}^{\text{curr}} \in \mathbb{R}^{T\times D}$ denote the embedded Siamese features. Different base models correspond to different $\text{Embed}(\cdot)$. Regarding PatchTST (Nie et al., 2023), the patch-wise embedding function $\text{PatchEmbed}(\cdot)$ is used to divide each variable into several patches and each patch is mapped to a patch token. As for iTransformer (Liu et al., 2024), it uses the variable-wise embedding $\text{VariateEmbed}(\cdot)$ and maps the entire variable into a temporal token. Note that lineage embeddings are used to identify the temporal distance between Siamese subseries. It is shared along the time dimension when being added to Siamese subseries features. Here, $\oplus$ represents the addition operation with the temporal dimension broadcast.

Next, TimeSiam utilizes Siamese encoders to process pairs of Siamese pair features, which can be instantiated as advanced time series models, e.g. PatchTST (Nie et al., 2023) or iTransformer (Liu et al., 2024). After the Siamese encoder layer, we can obtain pairs of representations of past and masked current subseries as follows:

$$\mathbf{h}_e^{\text{past}} = \text{Encoder}(\mathbf{z}^{\text{past}}),\ \widetilde{\mathbf{h}}_e^{\text{curr}} = \text{Encoder}(\widetilde{\mathbf{z}}^{\text{curr}}), \tag{3}$$

where $\mathbf{h}_e^{\text{past}}, \widetilde{\mathbf{h}}_e^{\text{curr}} \in \mathbb{R}^{T\times D}$ are from the Siamese encoder.

Note that our Siamese sampling strategy natively derives a past-to-current reconstruction task. As shown in Figure 2, we use a decoder that integrates cross-attention and self-attention mechanisms (Vaswani et al., 2017) to incorporate past information into the current subseries for reconstruction, which can inherently capture the temporal correlations. Besides, this design can also enrich the limited context of masked current series to ensure accurate reconstruction for representation learning. Concretely, $\widetilde{\mathbf{h}}_e^{\text{curr}}$ serves as the query, and $\mathbf{h}_e^{\text{past}}$ acts as both the key and value, generating the decoder representation of the current time subseries, denotes as $\widehat{\mathbf{h}}_d$. This representation undergoes further refinement through a self-attention layer and a Feed-Forward Network (FFN). We formalize the decoder process as follows:

$$\begin{aligned} \widehat{\mathbf{h}}_d &= \text{LayerNorm}\left(\widetilde{\mathbf{h}}_e^{\text{curr}} + \text{Cross-Attn}\left(\widetilde{\mathbf{h}}_e^{\text{curr}}, \mathbf{h}_e^{\text{past}}, \mathbf{h}_e^{\text{past}}\right)\right) \\ \mathbf{h}'_d &= \text{LayerNorm}\left(\widehat{\mathbf{h}}_d + \text{Self-Attn}\left(\widehat{\mathbf{h}}_d, \widehat{\mathbf{h}}_d, \widehat{\mathbf{h}}_d\right)\right) \\ \mathbf{h}_d &= \text{LayerNorm}\left(\mathbf{h}'_d + \text{FFN}\left(\mathbf{h}'_d\right)\right). \end{aligned} \tag{4}$$

We summarize this process as $\mathbf{h}_d = \text{Decoder}(\widetilde{\mathbf{h}}_e^{\text{curr}}, \mathbf{h}_e^{\text{past}})$. Finally, the output of the decoder $\mathbf{h}_d \in \mathbb{R}^{T\times D}$ is used to reconstruct the masked current subseries through a linear projection layer, which can be formalized as:

$$\widehat{\mathbf{x}}^{\text{curr}} = \text{Projector}(\mathbf{h}_d). \tag{5}$$

Benefiting from our design, TimeSiam can be supervised by a simple reconstruction loss function and inherently learn time-dependent representations by past-to-current temporal correlation modeling. The loss for each Siamese pair is

$$\mathcal{L}_{\text{reconstruction}} = \|\mathbf{x}^{\text{curr}} - \widehat{\mathbf{x}}^{\text{curr}}\|_2^2. \tag{6}$$

3.2. Fine-tuning

Under the cooperation of lineage embeddings, the pre-trained Siamese encoder can capture diverse temporal-related representations under different lineage embeddings. As demonstrated in Figure 2(b), this advantage can further derive two types of fine-tuning paradigms, covering both fixed and extended input series settings.

Fixed-Input-Multiple-Lineages In a standard fine-tuning scenario, a sample typically generates only one type of representation, which seriously limits the capacity of the pre-trained encoder. In contrast, TimeSiam innovatively pre-trains Siamese encoders with diverse lineage embeddings to capture different distanced temporal correlations, which allows TimeSiam to derive diverse representations with different lineages for the same input series. This procedure subtly releases the capacity of the pre-trained model and enhances the diversity of extracted representations. Given an input series $\mathbf{x} \in \mathbb{R}^{T \times C}$, this process can be written as

$$\begin{aligned} \overline{\mathbf{h}}_e &= \text{Average}\left(\mathbf{h}_{e,0}, \mathbf{h}_{e,1}, ...\mathbf{h}_{e,n}\right), \\ \text{where } \mathbf{h}_{e,i} &= \text{Encoder}\left(\text{Embed}(\mathbf{x}) \oplus \mathbf{e}_i^{\text{lineage}}\right). \end{aligned} \tag{7}$$

The final output $\overline{\mathbf{h}}_e \in \mathbb{R}^{T \times D}$ is an ensemble of a set of temporal representations derived from the same input series $\mathbf{x}$ but with different lineage embeddings $\mathbf{e}_i^{\text{lineage}}$, which can cover diverse temporal-related information of input series.

Extended-Input-Multiple-Lineages Note that in the fine-tuning stage, the model may receive longer records than the pre-training series. Given a $(k+1)T$-length input $(\mathbf{x}_k, \cdots, \mathbf{x}_1, \mathbf{x}_0), \mathbf{x}_i \in \mathbb{R}^{T \times C}$, previous time series pre-training methods have to adopt the same encoder to different segments, which clearly overlooks the chronological order of extended series. Desirably, in TimeSiam, we can leverage multiple lineage embeddings trained under different temporal distanced pairs to different segments, which can natively conserve the temporal order of different segments. This advantage is achieved by associating each segment with its respective lineage embedding:

$$\begin{aligned} \overline{\mathbf{h}}_e &= \text{Concat}\left(\mathbf{h}_{e,0}, \mathbf{h}_{e,1}, \ldots, \mathbf{h}_{e,k}\right), \\ \text{where } \mathbf{h}_{e,i} &= \text{Encoder}\left(\text{Embed}(\mathbf{x}_i) \oplus \mathbf{e}^{\text{lineage}}_{\text{LineageMatching}(iT)}\right). \end{aligned} \tag{8}$$

Here $\overline{\mathbf{h}}_e \in \mathbb{R}^{(k+1)T \times D}$ denotes the extracted representation for extended input series.

To align the experiment setting with previous work (Dong et al., 2023), our experiments are based on the Fixed-Input-Multiple-Lineages setting except for special clarification.

4. Experiments

We perform extensive experiments across two mainstream time series analysis tasks: forecasting and classification, covering both in- and cross-domain settings.

4.1. Experimental Setup

Datasets We summarize the experimental benchmarks in Table 1, encompassing eleven well-established datasets and two newly constructed datasets, which cover two primary tasks in time series analysis: forecasting and classification. It is worth noting that to further demonstrate the pre-training benefits under large and diverse data, we employ the TSLD dataset, which is constructed by merging time series datasets from multiple domains that are nonoverlapping with the other datasets. This allows us to explore cross-domain transfer scenarios with large-scale pre-training data. Please refer to Appendix B for a more comprehensive description.

Table 1. Summary of experiment benchmarks, where TSLD-500M and TSLD-1G are newly constructed from diverse domains.

TASKS	DATASETS	DOMAIN	EXAMPLES
Forecasting	ETT (4 subsets)	Electricity	14.3K
	Weather	Weather	52.7K
	Electricity	Electricity	26.3K
	Traffic	Transportation	17.5K
	Exchange	Finance	7.6K
	TSLD-500M	Multiple	412.6K
	TSLD-1G	Multiple	13.9M
Classification	AD	EEG	5.97K
	TDBrain	EEG	11.9K
	PTB	ECG	62.4K

Backbone We use the advanced time series models across various tasks as Siamese encoders to evaluate the efficacy of our pre-training methods. In particular, we utilized iTransformer (Liu et al., 2024) and PatchTST (Nie et al., 2023) as the encoder for time series forecasting following their original configurations. The patch length and stride were both set to 12 without any overlap. TCN (Bai et al., 2018) is used as the backbone for the classification task (Wang et al., 2023). Note that to ensure a fair comparison, we unify the encoder backbone of our model and all the baselines. The results with a unified encoder generally surpass the results reported by themselves across all baselines.

Baselines We compare our TimeSiam with eight advanced self-supervised time series pre-training baselines under the in-domain setting, including contrastive learning methods: COMET (2023), TF-C (2022), LaST (2022), CoST (2022), TS2Vec (2022), TNC (2021), CPC (2018) and masked modeling methods: SimMTM (2023), Ti-MAE (2023b), TST

Table 2. In-domain fine-tuning for time series forecasting. Siamese encoders are both pre-trained and fine-tuned on the same dataset. Results are the average Mean Squared Error (MSE), calculated from forecasts made for four future lengths $O \in \{96, 192, 336, 720\}$, based on the past 96 time points. A smaller MSE indicates a better prediction. Full results are presented in Appendix G.

Encoder	Method	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Exchange	ECL	Traffic
PatchTST	Random init.	0.473	0.385	0.390	0.285	0.259	0.367	0.216	0.490
	CPC (2018)	0.440	0.401	0.389	0.290	0.272	0.368	0.220	0.504
	TNC (2021)	0.445	0.379	0.386	0.287	0.270	0.362	0.212	0.501
	TS2Vec (2022)	0.456	0.376	0.393	0.289	0.256	0.363	0.199	0.472
	CoST (2022)	0.457	0.374	0.395	0.286	0.253	0.364	0.203	0.480
	LaST (2022)	0.479	0.385	0.398	0.285	0.252	0.433	0.207	0.520
	TF-C (2022)	0.453	0.378	0.389	0.281	0.257	0.362	0.202	0.487
	TST (2021)	0.452	0.383	0.380	0.288	0.259	0.385	0.197	0.486
	Ti-MAE (2023b)	0.448	0.379	0.384	0.279	0.257	0.370	0.196	0.481
	PatchTST† (2023)	0.442	0.381	0.379	0.285	0.267	0.358	0.200	0.484
	SimMTM (2023)	0.440	0.382	0.377	0.285	0.256	0.361	0.192	0.466
	TimeSiam	**0.429**	**0.373**	**0.374**	**0.279**	**0.252**	**0.353**	**0.189**	**0.453**
iTransformer	Random init.	0.454	0.383	0.407	0.288	0.258	0.365	0.178	0.428
	TS2Vec (2022)	0.474	0.379	0.411	0.290	0.264	0.364	0.246	0.485
	CoST (2022)	0.472	0.386	0.411	0.294	0.269	0.366	0.252	0.529
	LaST (2022)	0.465	0.386	0.400	0.302	0.262	0.386	0.237	0.477
	TF-C (2022)	0.450	0.379	0.403	0.292	0.265	0.372	0.222	0.432
	TST (2021)	0.447	0.376	0.399	0.291	0.261	0.363	0.228	0.438
	Ti-MAE (2023b)	0.448	0.378	0.399	0.289	0.257	0.366	0.217	0.430
	SimMTM (2023)	0.445	0.376	0.397	0.286	0.259	0.358	0.179	0.426
	TimeSiam	**0.440**	**0.371**	**0.390**	**0.284**	**0.256**	**0.355**	**0.175**	**0.420**

(2021), and a patch-wise self-supervised masked modeling method, PatchTST, proposed by (Nie et al., 2023).

It should be noted that some baselines such as CPC (Oord et al., 2018), TNC (Tonekaboni et al., 2021), etc. are not validated in the iTransformer backbone because their specific design based on internal modeling of series is not applicable to iTransformer (Liu et al., 2024), which models the entire sequence as a whole temporal token and focuses on modeling relationships between different variables. In the cross-domain setting, due to the large-scale dataset TSLD will bring huge experiment costs, we only select part of the baselines (the efficient ones) for comparisons. Besides, COMET (2023) is specifically designed for medical time series and we also exclude it from cross-domain evaluation. More implementation details can be found in Appendix A.

4.2. Main Results

As shown in Figure 3,we summarize the performance of our TimeSiam in both in- and cross-domain scenarios for two mainstream time series analysis tasks: time series forecasting (x-axis) and classification (y-axis). For each scenario, TimeSiam exhibits significant improvement over other established strong self-supervised baselines for time series.

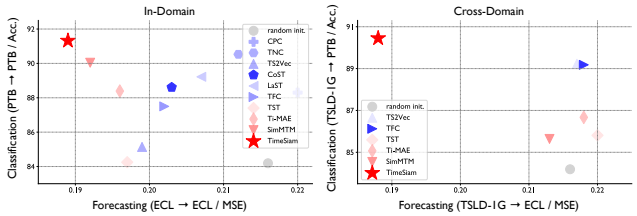

Figure 3. Comparison of time series pre-training baselines for forecasting (MSE ↓) and classification (Accuracy ↑) tasks. This comparison included both contrastive-based and masking-based methods, covering both in- (left) and cross-domain (right) settings.

4.3. Forecasting

In-domain We investigate the effectiveness of TimeSiam by integrating it with state-of-the-art time series forecasting models: PatchTST (Nie et al., 2023) and iTransformer (Liu et al., 2024). As shown in Table 2, TimeSiam can further enhance model performance, achieving an average MSE reduction of 5.7% and 2.5% across all forecasting benchmarks, even though these advanced models already exhibit excellent forecasting capabilities from random initialization. Significantly, it is evident that the masked modeling-based approach overall outperforms the contrastive-based approach in the forecasting task. This can be attributed

Table 3. Cross-domain fine-tuning for time series forecasting. Siamese encoders are pre-trained on the TSLD-1G dataset and fine-tuned on various target datasets. Results are the average Mean Squared Error (MSE) for four future lengths $O \in \{96, 192, 336, 720\}$, based on the past 96 time points. A smaller MSE indicates a better prediction. Full results are detailed in Appendix G.

Encoder	Method	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Exchange	ECL	Traffic
PatchTST	Random init.	0.473	0.385	0.390	0.285	0.259	0.367	0.216	0.490
	TS2Vec (2022)	0.441	0.375	0.380	0.291	0.256	0.365	0.217	0.528
	TF-C (2022)	0.437	0.378	0.386	**0.282**	0.264	0.360	0.218	0.543
	TST (2021)	0.434	0.384	0.387	0.303	0.263	0.365	0.220	0.514
	Ti-MAE (2023b)	0.435	0.374	0.380	0.294	0.256	0.362	0.218	0.515
	SimMTM (2023)	0.429	0.380	0.375	0.287	0.252	0.365	0.213	0.459
	TimeSiam	**0.425**	**0.374**	**0.371**	0.286	**0.251**	**0.360**	**0.188**	**0.454**

to the advantages gained through series reconstruction to learn low-level temporal representation. However, our TimeSiam still performs the best forecasting capability among all existing state-of-the-art self-supervised baseline methods.

Cross-domain As shown in Table 3, we use the TSLD-1G dataset, which contains larger scale time series samples from diverse domains, to validate the effectiveness of TimeSiam in a cross-domain transfer setting. Note that this setting not only requires pre-training learning from large-scale data but also poses thorny challenges in handling mismatched data distribution. The results consistently demonstrate that our TimeSiam significantly enhances the performance over training from random initialization covering all forecasting benchmarks, achieves comparable results in the in-domain setting, and consistently outperforms other baseline methods. It is worth noting that the transfer results even show superior performance compared to the in-domain scenario in some datasets, such as TSLD-1G $\rightarrow$ {ETTh1, ETTm1}. This confirms the essential significance of using more large-scale and varied data for time series pre-training.

4.4. Classification

In-domain To further investigate the generalizability of the representations learned by TimeSiam, we examined the impact of in-domain pre-training on classification tasks within the medical domain, following the setup in (Wang et al., 2023). Results in Table 4 demonstrate competitive outcomes achieved by both COMET (Wang et al., 2023) and SimMTM (Dong et al., 2023). This can be attributed to the elaborative designs of their approaches, where COMET incorporates domain-specific knowledge into its design and SimMTM models both high-level and low-level temporal representations. Compared with these competitive baselines, our TimeSiam, characterized by its simplicity and generality, consistently achieves remarkable results. In all classification benchmarks, our proposed TimeSiam consistently enhances the average classification accuracy by 11.5% compared to random initialization, surpassing other baseline methods.

Table 4. In-domain fine-tuning for time series classification. The model is pre-trained on two EEG datasets: AD and TDBrain, and an ECG dataset: PTB, and then fine-tuned on the same dataset. Accuracy (%) is recorded. See Appendix G for more details.

Method	AD	TDBrain	PTB
Random init.	80.62	79.08	84.19
CPC (2018)	77.40	85.19	88.30
TNC (2021)	78.58	85.21	90.53
TS2vec (2022)	81.26	80.21	85.14
CoST (2022)	73.87	83.86	88.61
LaST (2022)	72.63	85.13	89.22
TF-C (2022)	75.31	66.62	87.50
COMET (2023)	84.50	85.47	87.84
TST (2021)	81.50	83.22	84.25
Ti-MAE (2023b)	80.70	88.16	88.39
SimMTM (2023)	86.19	84.81	90.04
TimeSiam	**89.93**	**90.67**	**91.32**

Table 5. Cross-domain fine-tuning for time series classification. The model is pre-trained on TSLD-1G dataset and fine-tuned on EEG dataset AD, TDBrain and ECG dataset PTB. Accuracy (%) is recorded and further details can be found in Appendix G.

Method	AD	TDBrain	PTB
Random init.	80.62	79.08	84.19
TS2vec (2022)	80.59	85.58	89.23
TF-C (2022)	87.98	82.84	89.18
TST (2021)	82.60	83.65	85.81
Ti-MAE (2023b)	80.40	85.22	86.67
SimMTM (2023)	87.74	85.29	85.64
TimeSiam	**90.47**	**86.26**	**90.45**

Cross-domain Furtherly, we investigated the impact of using larger and more diverse pre-training datasets on time series classification tasks: TSLD-1G $\rightarrow$ {AD, TDBrain, PTB}.

Table 6. Ablations on the Traffic benchmark, which are conducted under the in-domain forecasting setting. The length of the input series was fixed at 96 time points. The default setting is indicated by a grey bold marking. MSE averaged from 4 forecasting horizons $\{96, 192, 336, 720\}$ is reported. The (c) notation in masking rules refers to channel-wise masking. More results can be found in Table 20.

(a) SIAMESE SAMPLING

Sampling range ($\max d$)		Lineage types N	
w/o	0.462	w/o	0.457
1	0.454	2	0.456
3	0.454	**3**	**0.453**
6	**0.453**	6	0.455
12	0.455		

(b) SUBSERIES RECONSTRUCTION

Masking rules		Masked ratio	
binomial	0.459	w/o	0.460
continuous	0.459	15%	0.459
mask last	0.461	**25%**	**0.453**
binomial (c)	0.457	50%	0.455
continuous (c)	**0.453**	75%	0.457

Generally, as shown in Table 5, pre-training will bring consistent promotion w.r.t. random initialization. However, it is also observed that employing larger datasets from more diverse domains does not definitively show an advantage over the in-domain pre-trained models. This result may come from the inherent differences between the domains of pre-training and fine-tuning under the cross-domain setting. Notably, even in this tough cross-domain setting, TimeSiam still surpasses the other baselines, demonstrating its capability to handle shifted data distributions.

4.5. Ablation Studies

We conduct extensive ablation studies to evaluate the effectiveness of various designs in TimeSiam, including Siamese sampling and subseries reconstruction.

Siamese Sampling We explore the key hyper-parameters of Siamese modeling: the maximum sampling distance between the past and current Siamese subseries (Sampling range $\max d$) and the size of the lineage embedding set (Lineage types N). The maximum sampling range is determined by the length of the subseries (T) and a hyperparameter (r), resulting in $\max d = T \times r$. Table 6(a) demonstrates that integrating past-to-current Siamese modeling outperforms self-reconstruction modeling in time series pre-training. Furthermore, expanding the sampling range of Siamese subseries reasonably significantly enhances performance, underscoring the critical role of Siamese modeling in achieving optimal fine-tuning results.

Subseries Reconstruction For subseries reconstruction, results in Table 6(b) indicate that channel-wise masking significantly benefits Siamese modeling, especially when comparing the *continuous* to channel-wise *continuous (c)*. Both random and continuous masking work well, achieving lower mean squared error in the process. As for the masking ratio, we find that a high masking ratio of 75% will significantly distort temporal variations, whereas a low mask ratio of 15% overly simplifies the task, hindering effective temporal representation learning. Therefore, we adopt a default masking ratio of 25%, same as (Devlin et al., 2018).

Table 7. Fine-tuning performance of TimeSiam under different pre-training data and model sizes. Relative improvement over random initialization (%) is marked in green. See Appendix A.3 for details.

PRE-TRAIN	TRAFFIC	ECL
Random initiation	0.490	0.216
TimeSiam in-domain	0.453	0.189
$\text{TimeSiam}_{\text{Base}}$ $\text{TSLD}_{\text{500M}}$	0.462 (+5.7)	0.189 (+12.5)
$\text{TimeSiam}_{\text{Base}}$ TSLD_{1G}	0.454 (+7.4)	0.188 (+12.5)
$\text{TimeSiam}_{\text{Large}}$ TSLD_{1G}	0.433 (+11.6)	0.185 (+14.4)

Lineage embeddings As shown in Table 8, we can observe that lineage embeddings play a vital role in enhancing forecasting performance. This comes from the inherent ability of lineage embeddings to distinguish temporal distances between subseries during the pre-training phase. As a result, they facilitate the learning of diverse temporal correlations. Consequently, the inclusion of lineage embeddings has been shown to be effective in improving performance when fine-tuning models for a variety of downstream tasks.

Table 8. Ablations on lineage embeddings. The MSE averaged from 4 forecasting horizons {96, 192, 336, 720} is reported here.

DATASETS	RANDOM INIT.	TIMESIAM	W/O LINEAGE
ETTh1	0.473	0.429	0.433 ↓
ETTm1	0.390	0.374	0.378 ↓
Weather	0.259	0.252	0.256 ↓
Traffic	0.490	0.453	0.457 ↓
Exchange	0.36	0.353	0.365 ↓

4.6. Analysis Experiment

Data Scale and Model Capacity One of the bottlenecks that block the development of time series pre-training is the lack of large-scale and diverse data for pre-training (Zhou et al., 2023). To investigate the influence of data scale on TimeSiam, we employed TimeSiam for pre-training on a larger dataset TSLD-{0.5G, 1G} along with different model sizes, followed by applying it to downstream time-series prediction tasks to assess fine-tuning effects. Results, illustrated

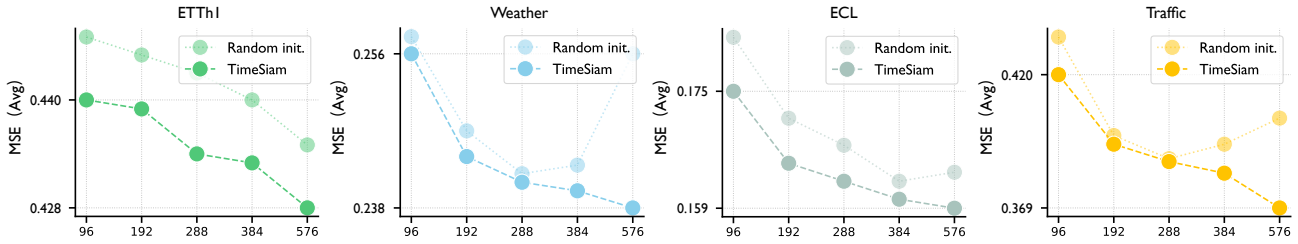

Figure 4. Fine-tuning the pre-trained model to the inputs with extended length {96, 192, 288, 384, 576} based on iTransformer (Liu et al., 2024). The MSE averaged from all predicted horizons {96, 192, 336, 720} is reported. Additional results are in the Appendix F.

in Table 7, reveal that the model performance is promoted significantly, empowering by TimeSiam pre-training (TimeSiam vs. random init.). Furthermore, although $\text{TSLD}_{500\text{M}}$ does not exhibit a significant advantage over TimeSiam under the in-domain setting initially, we observed a marked enhancement in performance as the dataset size increased ($\text{TSLD}_{1\text{G}}$ vs. $\text{TSLD}_{500\text{M}}$), and $\text{TimeSiam}_{\text{Large}}$ significantly outperforms $\text{TimeSiam}_{\text{Base}}$ in the $\text{TSLD}_{1\text{G}}$ finetuning scenarios, especially in the Traffic benchmark. This observation highlights the efficacy of TimeSiam and the positive correlation between data-model size and the final performance.

Adapt to Extended-Length Input As illustrated in Eq. (8), TimeSiam can natively adapt to longer inputs. Figure 4 shows that the standard prediction framework may degenerate under extended input length, which may be because of the noises in longer series. Contrarily, benefiting from an ingenious integration of Siamese modeling and lineage embeddings, TimeSiam achieves more accurate predictions, even when predicting from extended input series.

Linear Probing As an important finetuning setting, we also experiment with the linear probing, where we fix the pre-trained encoder and only finetune the newly added projector at the end of the model. Figure 5 illustrates that TimeSiam demonstrates superior performance compared to other baselines in terms of overall *linear probing* performance. Interestingly, by only fine-tuning the model head, the average forecasting performance across the four ETT subsets is already comparable with the results obtained through full fine-tuning, and significantly outperforms training from random initialization (*MSE*: 0.365 vs. 0.383). This finding further validates the effectiveness of TimeSiam in learning generalizable representations for various downstream tasks.

Embedding Effectiveness To elucidate the advantages of employing varying numbers of lineage embeddings within a fixed sampling range for prediction, as illustrated in Figure 6, our findings consistently demonstrate that the incorporation of lineage embeddings enhances prediction performance. Furthermore, augmenting the number of embeddings to encompass a greater extent of lineage within reasonable limits reinforces the efficacy of long-term prediction. Experimental results validate that lineage embeddings introduce more

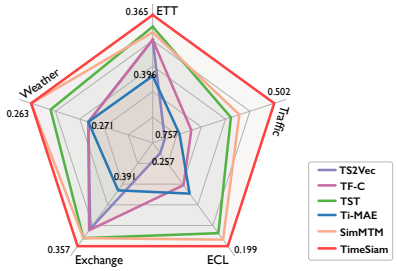

Figure 5. Linear probing on in-domain forecasting setting. Average results (MSE) are reported. Full results are shown in Table 17.

diverse temporal semantic information, enabling discrimination between different temporally distanced Siamese series, thereby boosting long-term prediction outcomes.

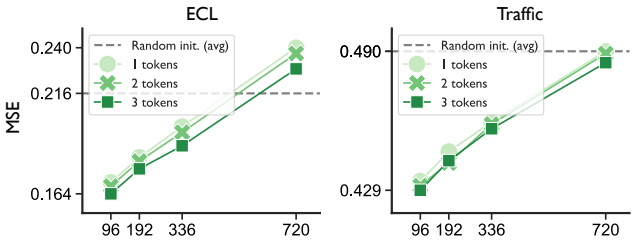

Figure 6. Increasing number of lineage embeddings on the ECL and Traffic. All results are under the "input-96" in-domain setting.

5. Conclusion

This paper proposes a simple, effective self-supervised pre-training framework named TimeSiam, uniquely focusing on temporal correlation modeling. TimeSiam employs Siamese networks as share encoders for randomly sampled past and current Siamese subseries. It further enhances data diversity through masking augmentation, which can also foster time-dependent representation learning by reconstructing current subseries from past observations. Additionally, we implement learnable lineage embeddings that efficiently capture disparities among Siamese subseries under different distances, enhancing the model's ability to cover diverse temporal correlations. Experimentally, TimeSiam demonstrated remarkable performance on various time series analysis tasks, consistently outperforming existing state-of-the-art baselines in both in- and cross-domain scenarios.

Acknowledgments

This work was supported by the National Key Research and Development Plan (2021YFB1715200), the National Natural Science Foundation of China (62022050 and U2342217), the BNRist Innovation Fund (BNR2024RC01010), and the National Engineering Research Center for Big Data Software. We are grateful to our colleagues Yipeng Huang and Zhiyao Cen for their support in the experimental efficiency.

Impact Statement

This paper focus on developing practical time series pre-training methods. We presents a novel approach based on Siamese networks, which could provide some inspiration for future research. The experimental results demonstrate the effectiveness of our approach across various domains and its potential value for real-world applications. It is essential to note that our work focuses solely on scientific issues, and we also ensure that ethical considerations are carefully taken into account. All the medical-related datasets are publicly available for scientific research. Thus, we believe that there is no ethical risk associated with our research.

References

Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. Signature verification using a" siamese" time delay neural network. In *NeurIPS*, 1993.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. *NeurIPS*, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *CVPR*, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2018.

Dong, J., Wu, H., Zhang, H., Zhang, L., Wang, J., and Long, M. Simmtm: A simple pre-training framework for masked time-series modeling. In *NeurIPS*, 2023.

Escudero, J., Abásolo, D., Hornero, R., Espino, P., and López, M. Analysis of electroencephalograms in alzheimer's disease patients with multiscale entropy. *Physiological measurement*, 27(11):1091, 2006.

Franceschi, J.-Y., Dieuleveut, A., and Jaggi, M. Unsupervised scalable representation learning for multivariate time series. In *NeurIPS*, 2019.

Gao, T., Fisch, A., and Chen, D. Making pre-trained language models better few-shot learners. In *IJCNLP*, 2020.

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, 2021.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220, 2000.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *SIGIR*, 2018.

Li, Y., Fan, H., Hu, R., Feichtenhofer, C., and He, K. Scaling language-image pre-training via masking. In *CVPR*, 2023a.

Li, Z., Rao, Z., Pan, L., Wang, P., and Xu, Z. Ti-mae: Self-supervised masked time series autoencoders. *arXiv preprint arXiv:2301.08871*, 2023b.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., and Tang, J. Self-supervised learning: Generative or contrastive. *TKDE*, 2021.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. itransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2024.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

PeMS. Traffic Dataset. http://pems.dot.ca.gov/.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. In *OpenAI blog*, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(1):5485–5551, 2020.

Tang, C. I., Perez-Pozuelo, I., Spathis, D., and Mascolo, C. Exploring contrastive learning in human activity recognition for healthcare. In *NeurIPS*, 2020.

Tonekaboni, S., Eytan, D., and Goldenberg, A. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

UCI. UCI Electricity Load Time Series Dataset. https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014.

Van Dijk, H., Van Wingen, G., Denys, D., Olbrich, S., Van Ruth, R., and Arns, M. The two decades brainclinics research archive for insights in neurophysiology (tdbrain) database. *Scientific data*, 9(1):333, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Wang, Y., Han, Y., Wang, H., and Zhang, X. Contrast everything: A hierarchical contrastive framework for medical time-series. In *NeurIPS*, 2023.

Wang, Z., Xu, X., Zhang, W., Trajcevski, G., Zhong, T., and Zhou, F. Learning latent seasonal-trend representations for time series forecasting. In *NeurIPS*, 2022.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., and Xu, H. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

Wetterstation. Weather Dataset. https://www.bgc-jena.mpg.de/wetter/.

Wickstrøm, K., Kampffmeyer, M., Mikalsen, K. Ø., and Jenssen, R. Mixing up contrastive learning: Self-supervised representation learning for time series. *PRL*, 2022.

Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *ICLR*, 2022.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, 2021.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023a.

Wu, H., Zhou, H., Long, M., and Wang, J. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 2023b.

Xiao, T., Wang, X., Efros, A. A., and Darrell, T. What should not be contrastive in contrastive learning. In *ICLR*, 2021.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022.

Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., and Xu, B. TS2Vec: Towards Universal Representation of Time Series. In *AAAI*, 2022.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., and Eickhoff, C. A transformer-based framework for multivariate time series representation learning. In *SIGKDD*, 2021.

Zhang, X., Zhao, Z., Tsiligkaridis, T., and Zitnik, M. Self-supervised contrastive pre-training for time series via time-frequency consistency. In *NeurIPS*, 2022.

Zhao, S., Jin, M., Hou, Z., Yang, C., Li, Z., Wen, Q., and Wang, Y. Himtm: Hierarchical multi-scale masked time series modeling for long-term forecasting. *arXiv preprint arXiv:2401.05012*, 2024.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.

Zhou, T., Niu, P., Wang, X., Sun, L., and Jin, R. One fits all: Power general time series analysis by pretrained lm. In *NeurIPS*, 2023.

A. Implementation Details

In this paper, all experiments were conducted on a single NVIDIA A100 SXM4 80GB GPU and implemented using the PyTorch framework (Paszke et al., 2019) for five repetitions. We evaluated performance using Mean Square Error (MSE) and Mean Absolute Error (MAE) for time series forecasting. For classification tasks, we comprehensively assessed model performance by measuring accuracy, precision, recall, F1 score, AUROC, and AUPRC.

A.1. Baseline Implementation

We have followed and compared the official implementations of all baselines to our approach. We have maintained the original configurations outlined in these papers to ensure a fair comparison. Note that we utilized an unofficial coding version of Ti-MAE (Li et al., 2023b) due to the unavailability of its official open-source implementation.

Table 9. Categories and open-source implementations of all baselines.

CATEGORIES	METHODS	OFFICIAL CODE LINK
Contrastive Learning	TS2VEC (Yue et al., 2022)	https://github.com/yuezhihan/ts2vec
	COST (Woo et al., 2022)	https://github.com/salesforce/CoST
	LAST (Wang et al., 2022)	https://github.com/zhycs/LaST
	TF-C (Zhang et al., 2022)	https://github.com/ mims-harvard/TFC-pretraining
	COMET (Wang et al., 2023)	https://github.com/DL4mHealth/COMET
Masked Modeling	TST (Zerveas et al., 2021)	https://github.com/gzerveas/mvts_transformer
	TI-MAE (Li et al., 2023b)	https://github.com/asmodaay/ti-mae
	PATCHTST (Nie et al., 2023)	https://github.com/yuqinie98/PatchTST
	SIMMTM (Xie et al., 2022)	https://github.com/thuml/simmtm

A.2. Training Configuration

We construct two types of pre-training and fine-tuning scenarios, in-domain and cross-domain, based on benchmarks for prediction and classification tasks to compare the effectiveness of our method with other time series pre-training methods. In the pre-training phase, we pre-train the model with different learning rates and batch sizes based on the pre-trained dataset. We then fine-tune it for downstream prediction and classification tasks supervised by L2 and Cross-Entropy loss, respectively. The configuration details are in Table A.3. Also, considering the size of the fine-tuned dataset and consistency with existing works, we fine-tune the model for 10 epochs for the prediction task and 50 epochs for the classification task.

Table 10. Pre-training and fine-tuning configurations in forecasting and classification tasks.

TASKS	PRE-TRAINING			FINE-TUNING			
	learning rate	batch size	epochs	learning rate	loss function	batch size	epochs
Forecasting	1e-4	32	50	1e-4	L2	{8, 16, 32}	10
Classification	1e-4	256	100	1e-4	Cross-Entropy	{32, 64, 128}	50

A.3. Model Configuration

We compare TimeSiam against eight state-of-the-art baselines for an unbiased and comprehensive comparison. To ensure the fairness of the evaluation, we choose state-of-the-art time series analysis models as a unified backbone for these pre-trained methods. Specifically, PatchTST (Nie et al., 2023) and iTransformer (Liu et al., 2024) are adopted for forecasting and employ Temporal Convolution Network (TCN) (Bai et al., 2018) for classification following the setup in (Wang et al., 2023).

In addition, we performed a hyperparameter search for all baselines, adhering to their official configuration in the in-domain setting. For Siamese encoders, we explored various configurations by adjusting the number of encoder layers (e_{layers}) and decoder layers (d_{layers}) from $\{1, 2, 3, 4\}$, selecting hidden dimensions (d_{model}) from $\{16, 32, 64, 128, 256, 512\}$ and attention heads (n_{heads}) from $\{8, 16, 32\}$. In the case of TCN models, we investigated different numbers of residual blocks, considering configurations of $\{5, 8, 10\}$. During the fine-tuning stage, we carefully consider the learning rate (lr) from $\{$1e-3, 5e-4, 1e-4, 1e-5$\}$, and head dropout (dropout) from $\{0, 0.1, 0.2, 0.3\}$ in order to enhance the adaptability of our pretrained model to diverse datasets.

Primarily, two model configurations with different sizes are explored in the cross-domain forecasting setting, that is **TimeSiam-Base** and **TimeSiam-Large**. These two models are used to evaluate the impact of model capacity on forecasting performance, specifically in the context of cross-domain pre-training and fine-tuning on large-scale data.

Table 11. Two experimental configurations of TimeSiam with different model sizes.

TYPES	CONFIGURATION					PARAMETERS
	e_{layers}	d_{layers}	d_{model}	d_{ff}	n_{heads}	
$\text{TimeSiam}_{\text{Base}}$	3	1	128	256	8	709,344
$\text{TimeSiam}_{\text{Large}}$	5	2	128	1024	16	2,554,720

B. Dataset Description

We conduct experiments on eleven well-established datasets and two newly constructed datasets covering two primary tasks in time series analysis: forecasting and classification. These datasets cover a variety of application scenarios, different types of signals, multivariate channel dimensions, varying time series lengths, large-span sampling frequencies, and different data sizes. The detailed descriptions of these datasets are summarized in Table 12.

B.1. Forecasting Datasets

(1) **ETT (4 subsets)** (Zhou et al., 2021) contains a group of four subsets oil temperature and power load collected by electricity transformers from July 2016 to July 2018 with minutes or hourly recorded frequence.

(2) **Weather** (Wetterstation) includes meteorological time series with 21 weather indicators collected every 10 minutes from the Weather Station of the Max Planck Biogeochemistry Institute in 2020.

(3) **Electricity** (UCI) records the hourly electricity consumption of 321 clients from 2012 to 2014.

(4) **Traffic** (PeMS) encompasses the hourly measures of road occupancy rates obtained from 862 sensors situated in the San Francisco Bay area freeways between January 2015 and December 2016.

(5) **Exchange** (Lai et al., 2018) records the daily exchange rates of eight different countries ranging from 1990 to 2016.

B.2. Classification Datasets

(1) **AD** (Escudero et al., 2006) has electroencephalography (EEG) recordings from 12 Alzheimer's patients and 11 healthy controls. Each patient has around 30 trials, each lasting for 5 seconds with 1280 timestamps (sampled at 256Hz) and includes 16 channels.

Table 12. Dataset descriptions. *Samples* are organized in (Train/Validation/Test).

TASKS	DATASETS	CHANNELS	SERIES LENGTH	SAMPLES	CLASSES	INFORMATION	FREQUENCY
Forecasting	ETTh1,ETTh2	7	{96,192,336,720}	8,545/2,881/2,881	-	Electricity	Hourly
	ETTm1,ETTm2	7	{96,192,336,720}	34,465/11,521/11,521	-	Electricity	15 Mins
	Weather	21	{96,192,336,720}	36,792/5,271/10,540	-	Weather	10 Mins
	Exchange	8	{96,192,336,720}	5,120/665/1,422	-	Exchange rate	Daily
	Electricity	321	{96,192,336,720}	18,317/2,633/5261	-	Electricity	Hourly
	Traffic	862	{96,192,336,720}	12,185/1,757/3,509	-	Transportation	Hourly
	TSLD-500M	1	{96,192,336,720}	369,030/31,872/-	-	Multi-domain	Mixing
	TSLD-1G	1	{96,192,336,720}	13,984,175/1,061,806/-	-	Multi-domain	Mixing
Classification	AD	16	256	4,329/891/747	3	EEG	256 Hz
	TDBrain	33	256	8,208/1,824/1,824	3	EEG	500 Hz
	PTB	15	300	53,950/3,400/5,020	3	ECG	1000 Hz

(2) **PTB** (Goldberger et al., 2000) has electrocardiogram (ECG) recordings from 290 patients with 15 channels sampled at 1000 Hz. This paper focuses on a subset of the dataset that includes 198 patients with heart diseases: Myocardial infarction and healthy controls.

(3) **TDBrain** (Van Dijk et al., 2022) monitors brain signals of 1274 patients with 33 channels during EC (Eye closed) and EO (Eye open) tasks. It includes 60 types of diseases, but this paper focuses on a subset of 25 Parkinson's disease patients and 25 healthy controls. Only the EC task trials are used for representation learning.

B.3. Merged Large Scale Datasets

To further substantiate the significance of time series pre-training on large-scale data and showcase its benefits in diverse and extensive datasets, we have amalgamated multiple non-overlapping time series datasets from various domains to construct the **T**ime **S**eries **L**arge **D**atasets (**TSLD**). In this paper, we present two versions of TSLD to valid our approach.

(1) **TSLD-500M** is a composite dataset comprising 400,902 samples from 12 time series datasets across the domains of Electricity, Transport, Energy, Climate, and others.

(2) **TSLD-1G**, building upon the TSLD-500M dataset, incorporates additional diverse datasets from domains such as Society, IoT, and Web. With an impressive sample count of 15,045,981 observations, TSLD-1G surpasses the size of datasets commonly used in time series analysis and provides greater diversity.

C. Masking Strategy

In this paper, we explored five different mask rules: binomial, channel binomial, continuous, channel continuous, and only masking the last to assess their impact on TimeSiam, illustrated in Figure 7.

(1) **Binomial masking**: Generate a mask by employing a binomial distribution across all channels within a given sample.

(2) **Channel binomial masking**: Generate a mask based on a binomial distribution that selectively masks individual channels at different timestamps within the sample.

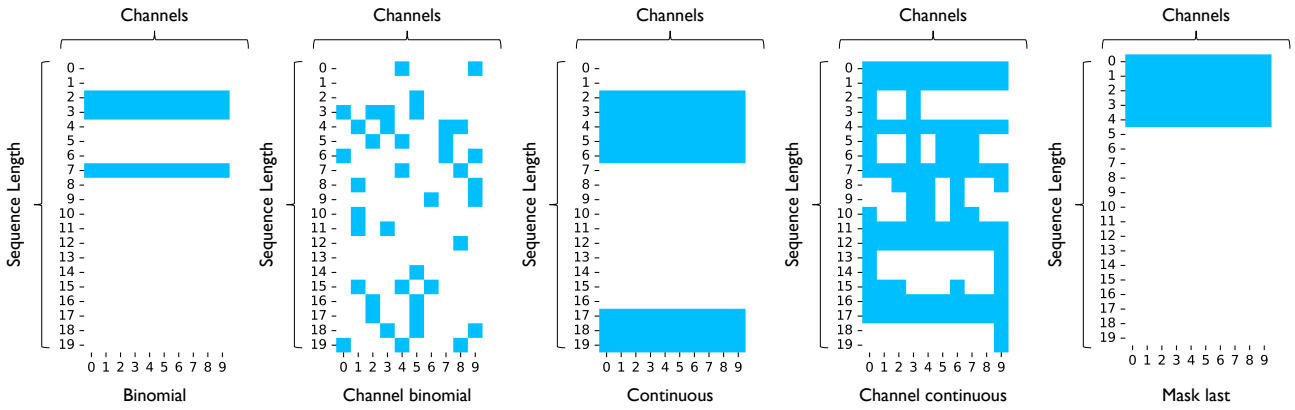

Figure 7. Showcases of various masking rules (75% masked ratio). The x-axis shows the channels and the y-axis represents the sequence length of the time series. Blue blocks indicate unmasked time stamps while white blocks represent masked ones.

(3) **Continuous masking**: Generate a mask by employing a geometric distribution across all channels within a given sample.

(4) **Channel continuous masking**: Generate a mask based on a geometric distribution that selectively masks individual channels at different timestamps within the sample.

(5) **Masking last**: Only mask the tail of time series in all channels.

D. Linear Probing and Full Fine-tuning

The results depicted in Figure 8 unequivocally demonstrate that both fine-tuning and linear probing methodologies utilizing TimeSiam outperform fully supervised learning from random initiation. Moreover, the findings suggest that full fine-tuning consistently yields superior results compared to linear probing across most datasets, with ETTh2 being a notable exception, where both approaches exhibit comparable performance.

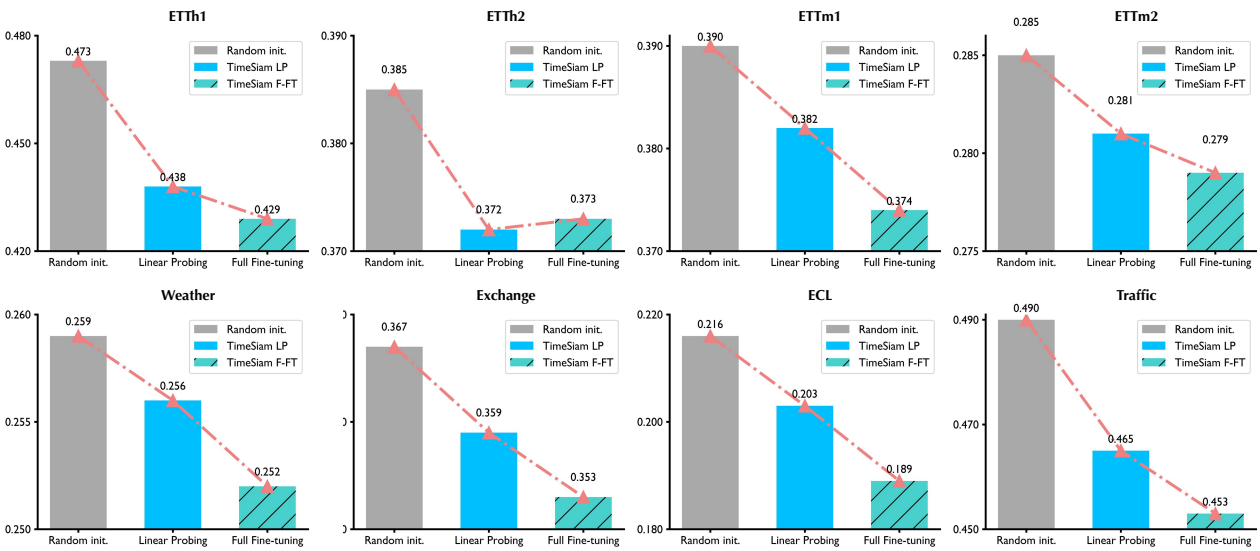

Figure 8. Comparison is made between the performance of linear probing pre-trained from TSLD-1G on various datasets and pre-training followed by fine-tuning on the same dataset. The mean squared error (MSE) is computed across all prediction lengths and serves as a measure of performance. A lower averaged MSE indicates superior predictive capability.

E. Multiple Lineages Representation Visualization

We employ Principal Components Analysis (PCA) to elucidate the distribution of temporal representations on the ECL dataset. We will only train the learned lineage embeddings during the pre-training phase. However, during downstream fine-tuning or linear probing, we will keep them fixed and not update them. It is worth noting that the embedded feature will be the same without different lineage embeddings. However, when time series is fed into a pre-trained Siamese network with different lineage embeddings, the model generates divergent temporal representations that representations derived from the same lineage embeddings tend to be closely clustered together, while representations from different lineage embeddings exhibit significant dissimilarity. Upon visual analysis, we have observed that the representations generated based on the same data but with different lineage embeddings exhibit a high level of diversity. This observation effectively validates the effectiveness of combining a pre-trained Siamese network with different lineage embeddings, which can enlarge the representation diversity.

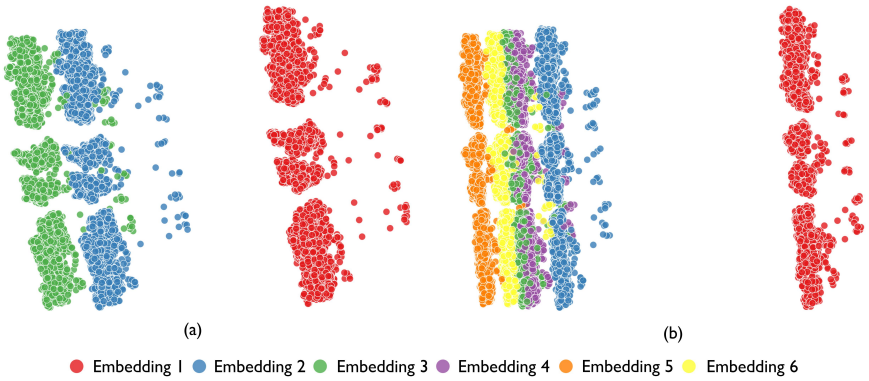

Figure 9. Visualizing the effect of temporal shift representations. (a) Visualization of test distribution under three types of lineage embeddings for ECL. (b) Visualization of test distribution under six types of lineage embeddings for ECL.

F. Adapt Extended Input Length

To facilitate the performance of Timesiam on fine-tuning scenarios with extended input lengths, we choose the input length to be an integral multiple of the pre-training length. In practice, the series length is not restricted to be an integral multiple of the pre-training length. TimeSiam can handle flexible input lengths, as different lineage embeddings can be shared across different time segments.

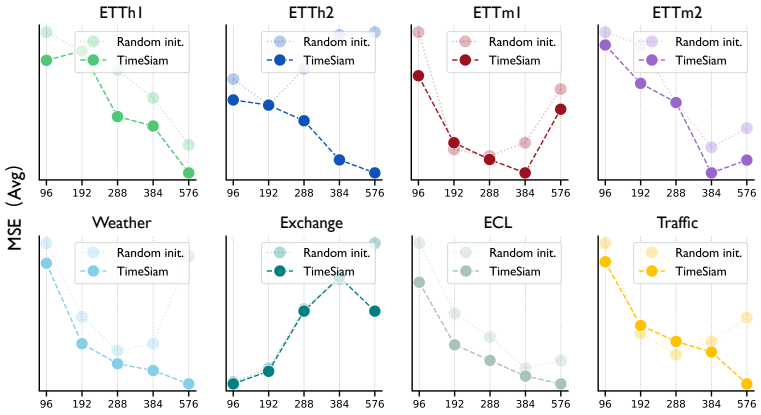

Figure 10. Full results for fine-tuning the pre-trained model with extended input length, where the input length is selected from {96, 192, 288, 384, 576}. The MSE averaged from four future lengths $O \in \{96, 192, 336, 720\}$ is reported.

G. Full Results

Due to the limited length of the text, we summarize the main experiments as follows:

Table 13. The main results for time series forecasting and classification tasks.

Experiments Categories	Tasks	Evaluation	Tabels Name
The main experiment	Forecasting	In-domain	Table 14, 15
		Cross-domain	Table 16
	Classification	In-domain	Table 18
		Cross-domain	Table 19

H. ShowCases

H.1. Different Masked Ratios

To investigate the reconstruction process of TimeSiam, we visually represent past time series, masked current time series, and reconstructed current time series with varying mask ratios using validation data from diverse datasets. Figure 11 demonstrates the reconstruction effects of TimeSiam at different mask ratios applied to the current time series. The context information is obtained by random sampling based on the current series, and the reconstruction becomes more challenging as the mask ratio increases due to the limited available information. Nevertheless, our TimeSiam model consistently achieves accurate reconstruction of masked current time series despite the scarcity of data and significant variation in temporal dimension between past and present. This accomplishment highlights the effectiveness of our approach in learning internal time-dependent representations through a past-to-current reconstruction.

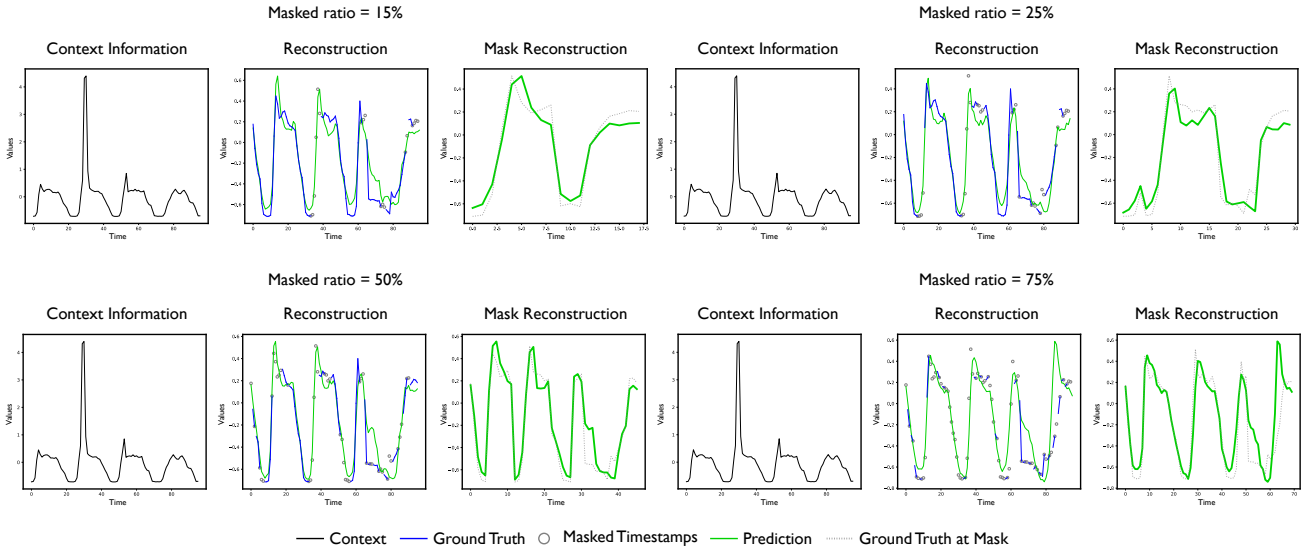

Figure 11. Showcases of TimeSiam in reconstructing time series with different masked ratios from Traffic.

H.2. Different Datasets

We further demonstrate the reconstruction effect across various datasets with different data distributions, as detailed below.

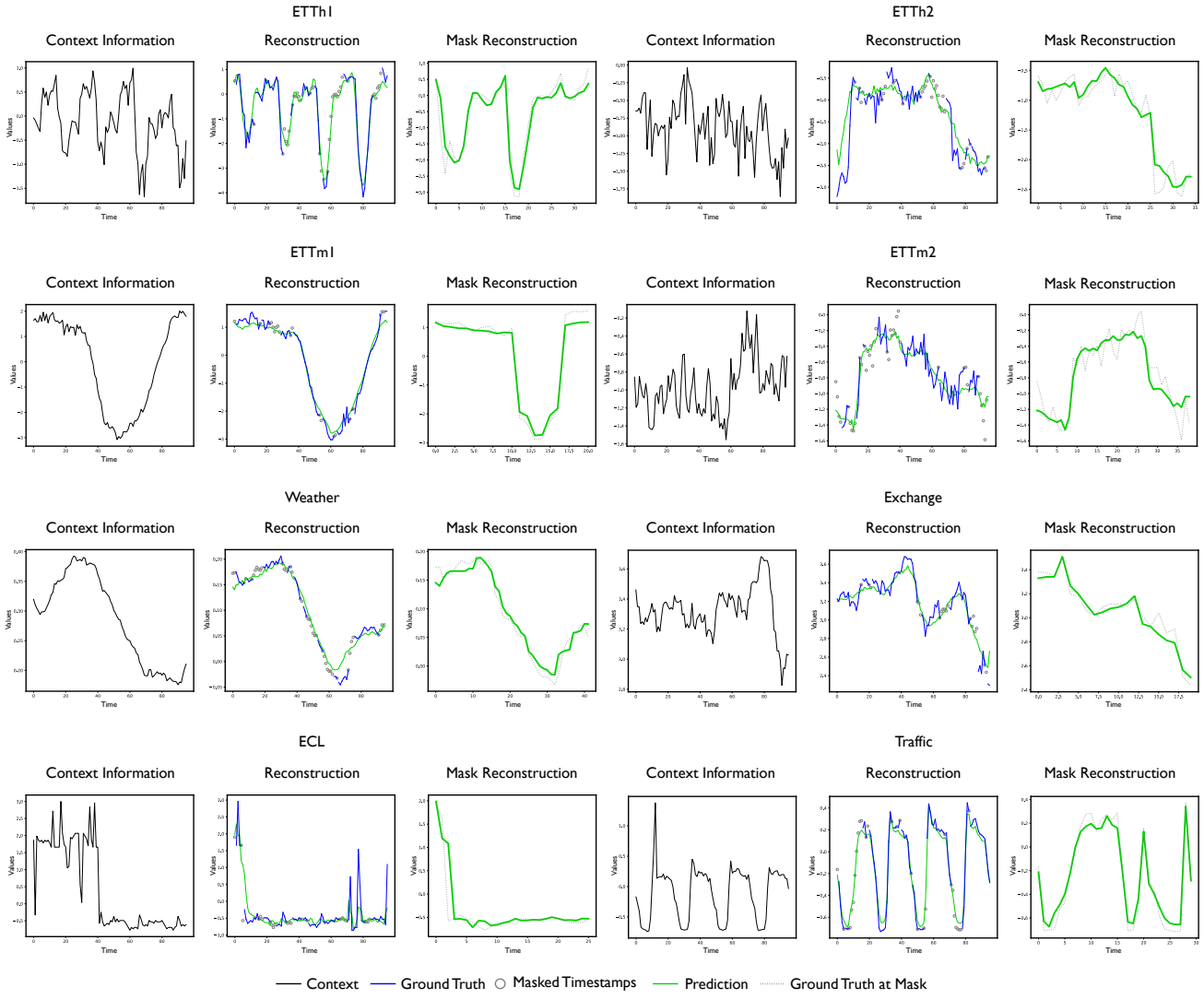

Figure 12. Showcases of TimeSiam in reconstructing time series from different datasets with 25% masked raito.

Table 14. Full results for the in-domain setting of forecasting using PatchTST. Pre-training and fine-tuning are performed on the same datasets. The standard deviations are within 0.005 for MSE and within 0.004 for MAE.

Methods		Random init.		CPC		TNC		TS2Vec		CoST		LaST		TFC		TST		Ti-MAE		SimMTM		**TimeSiame**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.420	0.423	0.380	0.401	0.377	0.397	0.381	0.400	0.383	0.405	0.396	0.413	0.399	0.420	0.377	0.401	0.396	0.415	0.367	0.389	0.378	0.401
	192	0.465	0.449	0.426	0.429	0.423	0.427	0.421	0.427	0.434	0.437	0.457	0.451	0.444	0.449	0.432	0.436	0.440	0.443	0.424	0.423	0.422	0.430
	336	0.504	0.470	0.465	0.451	0.471	0.453	0.468	0.452	0.474	0.460	0.507	0.478	0.479	0.467	0.475	0.461	0.481	0.462	0.473	0.456	0.459	0.452
	720	0.502	0.492	0.488	0.479	0.508	0.485	0.553	0.507	0.535	0.509	0.516	0.508	0.491	0.490	0.525	0.500	0.475	0.481	0.494	0.493	0.459	0.466
	Avg	0.473	0.458	0.440	0.440	0.445	0.441	0.456	0.447	0.457	0.453	0.469	0.463	0.453	0.457	0.452	0.450	0.448	0.450	0.440	0.440	**0.429**	**0.437**
ETTh2	96	0.297	0.345	0.313	0.364	0.291	0.343	0.297	0.343	0.288	0.342	0.294	0.345	0.302	0.345	0.304	0.358	0.300	0.353	0.299	0.352	0.293	0.345
	192	0.388	0.400	0.392	0.412	0.366	0.393	0.366	0.392	0.367	0.392	0.379	0.395	0.369	0.392	0.379	0.403	0.372	0.396	0.380	0.398	0.370	0.392
	336	0.426	0.434	0.438	0.449	0.427	0.438	0.416	0.430	0.413	0.429	0.423	0.436	0.412	0.428	0.412	0.432	0.418	0.431	0.422	0.432	0.410	0.424
	720	0.431	0.446	0.460	0.470	0.433	0.451	0.424	0.447	0.428	0.446	0.445	0.460	0.428	0.446	0.438	0.457	0.427	0.447	0.428	0.449	0.418	0.440
	Avg	0.385	0.406	0.401	0.424	0.379	0.406	0.376	0.403	0.374	0.402	0.385	0.409	0.378	0.403	0.383	0.413	0.379	0.407	0.382	0.408	**0.373**	**0.400**
ETTm1	96	0.330	0.368	0.324	0.361	0.323	0.361	0.325	0.364	0.348	0.377	0.345	0.381	0.353	0.378	0.319	0.360	0.325	0.363	0.317	0.356	0.319	0.360
	192	0.369	0.385	0.368	0.382	0.366	0.383	0.370	0.389	0.367	0.387	0.372	0.391	0.361	0.384	0.360	0.387	0.363	0.385	0.362	0.387	0.353	0.379
	336	0.400	0.407	0.403	0.405	0.399	0.405	0.405	0.415	0.404	0.414	0.412	0.420	0.392	0.406	0.391	0.408	0.396	0.409	0.387	0.405	0.383	0.402
	720	0.460	0.439	0.461	0.439	0.457	0.439	0.471	0.452	0.460	0.447	0.462	0.448	0.448	0.440	0.449	0.445	0.452	0.438	0.443	0.438	0.440	0.436
	Avg	0.390	0.400	0.389	0.397	0.386	0.397	0.393	0.405	0.395	0.406	0.398	0.410	0.389	0.402	0.380	0.400	0.384	0.399	0.377	0.397	**0.374**	**0.394**
ETTm2	96	0.175	0.258	0.196	0.281	0.187	0.261	0.174	0.261	0.181	0.269	0.177	0.258	0.281	0.327	0.181	0.265	0.175	0.261	0.175	0.262	0.175	0.261
	192	0.247	0.307	0.261	0.323	0.241	0.302	0.247	0.306	0.247	0.312	0.252	0.309	0.241	0.302	0.247	0.309	0.241	0.303	0.244	0.307	0.241	0.303
	336	0.309	0.345	0.302	0.343	0.313	0.363	0.306	0.345	0.309	0.348	0.307	0.344	0.304	0.343	0.314	0.354	0.301	0.341	0.312	0.351	0.300	0.341
	720	0.408	0.403	0.399	0.397	0.408	0.407	0.427	0.415	0.408	0.406	0.404	0.402	0.404	0.403	0.408	0.407	0.398	0.397	0.410	0.408	0.399	0.398
	Avg	0.285	0.328	0.290	0.336	0.287	0.333	0.289	0.332	0.286	0.334	0.285	0.328	0.281	0.327	0.288	0.334	0.279	0.326	0.285	0.332	**0.279**	**0.326**
Weather	96	0.177	0.218	0.193	0.230	0.191	0.231	0.174	0.216	0.171	0.214	0.170	0.212	0.177	0.218	0.177	0.221	0.175	0.218	0.184	0.220	0.171	0.213
	192	0.225	0.259	0.238	0.267	0.237	0.267	0.220	0.257	0.218	0.255	0.215	0.253	0.222	0.257	0.223	0.260	0.222	0.256	0.217	0.255	0.217	0.253
	336	0.278	0.297	0.292	0.306	0.292	0.305	0.276	0.297	0.273	0.295	0.272	0.295	0.277	0.296	0.279	0.301	0.278	0.299	0.273	0.296	0.272	0.293
	720	0.354	0.348	0.364	0.352	0.361	0.349	0.352	0.346	0.350	0.344	0.349	0.344	0.353	0.346	0.355	0.350	0.353	0.346	0.348	0.344	0.348	0.343
	Avg	0.259	0.281	0.272	0.289	0.270	0.288	0.256	0.279	0.253	0.277	0.252	0.276	0.257	0.279	0.259	0.283	0.257	0.280	0.256	0.279	**0.252**	**0.276**
Exchange	96	0.084	0.201	0.085	0.197	0.086	0.203	0.084	0.201	0.090	0.208	0.096	0.220	0.083	0.201	0.098	0.218	0.083	0.200	0.083	0.202	0.084	0.203
	192	0.187	0.307	0.187	0.308	0.180	0.301	0.185	0.306	0.179	0.301	0.190	0.313	0.173	0.296	0.187	0.308	0.186	0.307	0.182	0.303	0.176	0.300
	336	0.337	0.422	0.332	0.422	0.329	0.416	0.328	0.415	0.332	0.416	0.409	0.455	0.332	0.418	0.330	0.418	0.327	0.415	0.346	0.427	0.310	0.404
	720	0.858	0.695	0.867	0.694	0.851	0.694	0.856	0.696	0.854	0.698	1.035	0.749	0.860	0.698	0.925	0.731	0.882	0.708	0.831	0.689	0.842	0.690
	Avg	0.367	0.406	0.368	0.405	0.362	0.404	0.363	0.405	0.364	0.406	0.433	0.434	0.362	0.403	0.385	0.419	0.370	0.408	0.361	0.405	**0.353**	**0.399**
ECL	96	0.193	0.291	0.190	0.287	0.190	0.277	0.175	0.268	0.178	0.269	0.183	0.275	0.171	0.263	0.171	0.267	0.181	0.271	0.164	0.255	0.164	0.245
	192	0.199	0.297	0.204	0.290	0.194	0.283	0.183	0.275	0.185	0.275	0.190	0.281	0.188	0.277	0.181	0.276	0.197	0.277	0.178	0.268	0.173	0.256
	336	0.216	0.312	0.227	0.300	0.211	0.299	0.199	0.292	0.202	0.292	0.205	0.296	0.205	0.291	0.197	0.291	0.200	0.293	0.190	0.280	0.189	0.275
	720	0.257	0.345	0.257	0.347	0.254	0.334	0.240	0.324	0.245	0.326	0.248	0.330	0.244	0.322	0.237	0.325	0.205	0.326	0.235	0.318	0.229	0.310
	Avg	0.216	0.311	0.220	0.306	0.212	0.298	0.199	0.290	0.203	0.291	0.207	0.296	0.202	0.288	0.197	0.290	0.196	0.292	0.192	0.280	**0.189**	**0.272**
Traffic	96	0.472	0.305	0.449	0.487	0.483	0.309	0.309	0.291	0.458	0.294	0.506	0.330	0.465	0.301	0.478	0.292	0.463	0.295	0.442	0.285	0.429	0.279
	192	0.474	0.304	0.505	0.315	0.495	0.311	0.457	0.293	0.465	0.297	0.503	0.326	0.470	0.311	0.469	0.316	0.470	0.299	0.452	0.305	0.442	0.282
	336	0.491	0.331	0.514	0.343	0.504	0.333	0.474	0.301	0.480	0.304	0.517	0.332	0.498	0.320	0.482	0.323	0.486	0.321	0.473	0.322	0.456	0.288
	720	0.523	0.327	0.511	0.351	0.521	0.341	0.509	0.319	0.515	0.321	0.552	0.349	0.514	0.326	0.516	0.327	0.504	0.337	0.497	0.331	0.486	0.307
	Avg	0.490	0.317	0.504	0.330	0.501	0.324	0.472	0.301	0.480	0.304	0.520	0.334	0.487	0.315	0.486	0.315	0.481	0.313	0.466	0.311	**0.453**	**0.289**

Table 15. Full results for the in-domain setting of forecasting using iTransformer. Pre-training and fine-tuning are performed on the same datasets. The standard deviations are within 0.005 for MSE and within 0.004 for MAE.

Methods		Random init.		TS2Vec		CoST		LaST		TFC		TST		Ti-MAE		SimMTM		**TimeSiame**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.386	0.405	0.404	0.417	0.404	0.416	0.397	0.414	0.384	0.402	0.383	0.402	0.384	0.400	0.382	0.397	0.378	0.399
	192	0.441	0.436	0.459	0.448	0.457	0.447	0.449	0.444	0.439	0.432	0.436	0.433	0.436	0.430	0.435	0.427	0.429	0.428
	336	0.487	0.458	0.502	0.427	0.503	0.472	0.493	0.468	0.482	0.454	0.478	0.456	0.481	0.456	0.477	0.450	0.471	0.451
	720	0.503	0.491	0.531	0.509	0.523	0.502	0.520	0.501	0.496	0.484	0.492	0.489	0.492	0.486	0.485	0.476	0.483	0.481
	Avg	0.454	0.447	0.474	0.462	0.472	0.459	0.465	0.457	0.450	0.443	0.447	0.445	0.448	0.443	0.445	**0.438**	**0.440**	0.440
ETTh2	96	0.297	0.349	0.299	0.350	0.298	0.348	0.303	0.354	0.301	0.351	0.293	0.347	0.298	0.349	0.295	0.348	0.289	0.342
	192	0.380	0.400	0.384	0.403	0.382	0.400	0.381	0.402	0.377	0.396	0.375	0.398	0.378	0.397	0.374	0.397	0.367	0.390
	336	0.428	0.432	0.415	0.428	0.425	0.436	0.423	0.434	0.418	0.430	0.413	0.430	0.414	0.429	0.413	0.430	0.408	0.422
	720	0.427	0.445	0.419	0.442	0.440	0.454	0.438	0.451	0.420	0.441	0.421	0.443	0.420	0.441	0.420	0.443	0.418	0.440
	Avg	0.383	0.407	0.379	0.406	0.386	0.410	0.386	0.410	0.379	0.405	0.376	0.405	0.378	0.404	0.376	0.405	**0.371**	**0.398**
ETTm1	96	0.334	0.368	0.349	0.379	0.348	0.378	0.333	0.369	0.335	0.368	0.334	0.373	0.339	0.374	0.327	0.364	0.329	0.366
	192	0.377	0.391	0.387	0.396	0.388	0.397	0.377	0.392	0.378	0.390	0.377	00.396	0.375	0.391	0.372	0.386	0.368	0.386
	336	0.426	0.420	0.422	0.418	0.422	0.418	0.413	0.414	0.413	0.412	0.410	0.417	0.410	0.412	0.410	0.410	0.403	0.408
	720	0.491	0.459	0.485	0.454	0.487	0.455	0.477	0.451	0.485	0.452	0.475	0.455	0.471	0.446	0.478	0.450	0.466	0.445
	Avg	0.407	0.410	0.411	0.412	0.411	0.412	0.400	0.407	0.403	0.406	0.399	0.410	0.399	0.406	0.397	0.403	**0.392**	**0.401**
ETTm2	96	0.180	0.264	0.186	0.272	0.191	0.273	0.184	0.268	0.180	0.264	0.186	0.270	0.182	0.265	0.180	0.264	0.179	0.263
	192	0.250	0.309	0.251	0.312	0.254	0.312	0.268	0.317	0.251	0.308	0.252	0.312	0.248	0.308	0.246	0.307	0.245	0.306
	336	0.311	0.348	0.313	0.351	0.315	0.350	0.327	0.359	0.320	0.352	0.313	0.350	0.310	0.348	0.307	0.347	0.306	0.345
	720	0.412	0.407	0.409	0.403	0.416	0.407	0.427	0.414	0.416	0.407	0.411	0.407	0.415	0.407	0.409	0.405	0.405	0.401
	Avg	0.288	0.332	0.290	0.335	0.294	0.336	0.302	0.340	0.292	0.333	0.291	0.335	0.289	0.332	0.286	0.331	**0.284**	**0.329**
Weather	96	0.174	0.214	0.183	0.226	0.189	0.231	0.180	0.223	0.173	0.213	0.179	0.221	0.173	0.212	0.173	0.212	0.174	0.217
	192	0.221	0.254	0.231	0.266	0.236	0.269	0.228	0.262	0.222	0.257	0.225	0.261	0.221	0.256	0.225	0.258	0.222	0.256
	336	0.278	0.296	0.284	0.303	0.288	0.306	0.282	0.301	0.289	0.298	0.282	0.303	0.278	0.297	0.278	0.298	0.275	0.295
	720	0.358	0.349	0.359	0.351	0.363	0.354	0.358	0.350	0.377	0.350	0.358	0.352	0.357	0.349	0.359	0.349	0.350	0.346
	Avg	0.258	**0.278**	0.264	0.287	0.269	0.290	0.262	0.284	0.265	0.280	0.261	0.284	0.257	0.279	0.259	0.279	**0.255**	0.279
Exchange	96	0.086	0.206	0.088	0.208	0.090	0.213	0.091	0.212	0.087	0.208	0.090	0.211	0.087	0.208	0.087	0.211	0.092	0.215
	192	0.177	0.299	0.180	0.302	0.183	0.306	0.187	0.310	0.176	0.300	0.182	0.305	0.180	0.303	0.182	0.304	0.182	0.306
	336	0.331	0.417	0.332	0.418	0.335	0.420	0.333	0.421	0.347	0.428	0.332	0.419	0.333	0.418	0.330	0.411	0.341	0.426
	720	0.847	0.691	0.854	0.697	0.856	0.699	0.933	0.736	0.877	0.709	0.849	0.699	0.865	0.703	0.833	0.669	0.805	0.679
	Avg	0.360	0.403	0.364	0.406	0.366	0.410	0.386	0.420	0.372	0.411	0.363	0.409	0.366	0.408	0.358	**0.399**	**0.355**	0.407
ECL	96	0.148	0.240	0.214	0.310	0.225	0.318	0.202	0.296	0.191	0.278	0.196	0.292	0.185	0.281	0.145	0.236	0.147	0.239
	192	0.162	0.253	0.228	0.324	0.234	0.328	0.217	0.312	0.202	0.291	0.208	0.304	0.197	0.293	0.169	0.259	0.162	0.253
	336	0.178	0.269	0.247	0.340	0.253	0.344	0.239	0.331	0.222	0.310	0.230	0.323	0.219	0.312	0.176	0.267	0.175	0.269
	720	0.225	0.317	0.294	0.375	0.297	0.376	0.288	0.368	0.267	0.346	0.276	0.358	0.265	0.347	0.225	0.310	0.215	0.304
	Avg	0.178	0.270	0.246	0.337	0.252	0.342	0.237	0.327	0.222	0.306	0.228	0.319	0.217	0.308	0.179	0.268	**0.175**	**0.266**
Traffic	96	0.395	0.268	0.450	0.313	0.504	0.352	0.439	0.304	0.389	0.286	0.394	0.282	0.398	0.280	0.400	0.273	0.386	0.262
	192	0.417	0.276	0.469	0.321	0.509	0.352	0.462	0.315	0.398	0.297	0.403	0.301	0.405	0.294	0.412	0.280	0.411	0.272
	336	0.433	0.283	0.491	0.331	0.529	0.362	0.484	0.325	0.435	0.314	0.440	0.310	0.433	0.304	0.426	0.288	0.425	0.278
	720	0.467	0.302	0.531	0.352	0.572	0.383	0.523	0.347	0.504	0.344	0.514	0.343	0.483	0.333	0.466	0.307	0.458	0.297
	Avg	0.428	0.282	0.485	0.329	0.529	0.362	0.477	0.323	0.432	0.310	0.438	0.309	0.430	0.303	0.426	0.287	**0.420**	**0.277**

Table 16. Full results for the cross-domain setting of forecasting using PatchTST. Pre-training on the TSLD-1G dataset and fine-tune it on various target dataset. The standard deviations are within 0.005 for MSE and within 0.004 for MAE.

Methods		Random init.		TS2Vec		TFC		TST		Ti-MAE		SimMTM		**TimeSiame**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.420	0.423	0.384	0.407	0.384	0.405	0.391	0.412	0.384	0.405	0.376	0.402	0.371	0.398
	192	0.465	0.449	0.436	0.439	0.426	0.432	0.433	0.436	0.437	0.437	0.421	0.432	0.417	0.427
	336	0.504	0.470	0.462	0.456	0.463	0.453	0.458	0.450	0.462	0.454	0.454	0.455	0.448	0.447
	720	0.502	0.492	0.481	0.483	0.473	0.476	0.455	0.469	0.458	0.469	0.466	0.479	0.463	0.473
	Avg	0.473	0.458	0.441	0.446	0.437	0.442	0.434	0.442	0.435	0.441	0.429	0.442	**0.425**	**0.436**
ETTh2	96	0.297	0.345	0.295	0.347	0.296	0.347	0.298	0.346	0.287	0..342	0.293	0.347	0.292	0.345
	192	0.388	0.400	0.366	0.392	0.368	0.392	0.383	0.401	0.367	0.391	0.387	0.409	0.370	0.394
	336	0.426	0.434	0.413	0.426	0.421	0.433	0.428	0.435	0.409	0.425	0.421	0.428	0.410	0.427
	720	0.431	0.446	0.427	0.450	0.426	0.445	0.427	0.446	0.424	0.444	0.418	0.443	0.423	0.444
	Avg	0.385	0.406	0.375	0.404	0.378	0.404	0.384	0.407	**0.374**	**0.403**	0.380	0.407	**0.374**	**0.403**
ETTm1	96	0.330	0.368	0.320	0.360	0.325	0.363	0.326	0.364	0.319	0.360	0.316	0.357	0.309	0.352
	192	0.369	0.385	0.359	0.382	0.366	0.385	0.369	0.385	0.358	0.383	0.355	0.380	0.350	0.378
	336	0.400	0.407	0.395	0.407	0.398	0.408	0.399	0.407	0.398	0.411	0.386	0.400	0.383	0.402
	720	0.460	0.439	0.446	0.435	0.455	0.440	0.454	0.438	0.398	0.411	0.443	0.436	0.442	0.437
	Avg	0.390	0.400	0.380	0.396	0.386	0.399	0.387	0.399	0.380	0.398	0.375	0.393	**0.371**	**0.392**
ETTm2	96	0.175	0.258	0.176	0.261	0.176	0.259	0.188	0.271	0.179	0.265	0.177	0.264	0.182	0.268
	192	0.247	0.307	0.245	0.306	0.243	0.303	0.258	0.318	0.256	0.316	0.247	0.308	0.243	0.311
	336	0.309	0.345	0.310	0.347	0.303	0.342	0.334	0.361	0.325	0.359	0.309	0.348	0.314	0.351
	720	0.408	0.403	0.432	0.419	0.407	0.403	0.431	0.417	0.415	0.406	0.416	0.412	0.406	0.405
	Avg	0.285	0.328	0.291	0.333	**0.282**	**0.327**	0.303	0.342	0.294	0.337	0.287	0.333	0.286	0.334
Weather	96	0.177	0.218	0.174	0.216	0.184	0.222	0.188	0.231	0.175	0.216	0.170	0.214	0.170	0.214
	192	0.225	0.259	0.220	0.256	0.229	0.261	0.229	0.266	0.220	0.256	0.217	0.254	0.217	0.255
	336	0.278	0.297	0.277	0.297	0.284	0.300	0.281	0.303	0.276	0.296	0.273	0.295	0.270	0.295
	720	0.354	0.348	0.352	0.346	0.360	0.348	0.355	0.350	0.351	0.345	0.349	0.344	0.348	0.345
	Avg	0.259	0.281	0.256	0.279	0.264	0.283	0.263	0.288	0.256	0.278	0.252	**0.277**	**0.251**	**0.277**
Exchange	96	0.084	0.201	0.083	0.201	0.082	0.200	0.084	0.202	0.085	0.203	0.090	0.209	0.086	0.204
	192	0.187	0.307	0.176	0.298	0.174	0.297	0.178	0.299	0.179	0.302	0.171	0.297	0.179	0.301
	336	0.337	0.422	0.332	0.416	0.330	0.416	0.332	0.417	0.331	0.417	0.335	0.419	0.329	0.416
	720	0.858	0.695	0.867	0.700	0.853	0.696	0.867	0.698	0.851	0.696	0.862	0.690	0.849	0.694
	Avg	0.367	0.406	0.365	0.404	**0.360**	**0.402**	0.365	0.404	0.362	0.405	0.365	0.404	**0.360**	0.404
ECL	96	0.193	0.291	0.194	0.285	0.196	0.285	0.200	0.289	0.195	0.288	0.189	0.283	0.162	0.249
	192	0.199	0.297	0.199	0.291	0.200	0.291	0.202	0.292	0.201	0.294	0.196	0.289	0.172	0.259
	336	0.216	0.312	0.216	0.307	0.216	0.306	0.218	0.307	0.217	0.309	0.212	0.304	0.189	0.276
	720	0.257	0.345	0.257	0.339	0.258	0.339	0.259	0.338	0.258	0.341	0.254	0.337	0.227	0.309
	Avg	0.216	0.331	0.257	0.339	0.218	0.305	0.220	0.307	0.218	0.308	0.213	0.303	**0.188**	**0.273**
Traffic	96	0.472	0.305	0.513	0.340	0.532	0.354	0.514	0.329	0.499	0.328	0.437	0.280	0.430	0.277
	192	0.474	0.304	0.512	0.338	0.526	0.349	0.497	0.328	0.500	0.328	0.447	0.283	0.443	0.280
	336	0.491	0.331	0.525	0.342	0.539	0.354	0.504	0.328	0.512	0.332	0.460	0.289	0.456	0.286
	720	0.523	0.327	0.560	0.359	0.576	0.372	0.540	0.335	0.547	0.350	0.492	0.307	0.488	0.304
	Avg	0.490	0.317	0.528	0.345	0.543	0.357	0.514	0.330	0.515	0.335	0.459	0.290	**0.454**	**0.287**

Table 17. Full results for the in-domain setting of forecasting based on PatchTST. Pre-training and linear probing on the same dataset. The standard deviations are within 0.005 for MSE and within 0.004 for MAE. Note that the *Random init.* here refers to the train-from-scratch model, which will optimize the whole model. Thus, the *Random init.* is in the different setting w.r.t. other pre-training methods, where the latters are from linear probing.

Methods		Random init.		TS2Vec		TFC		TST		Ti-MAE		SimMTM		**TimeSiame**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.420	0.423	0.390	0.399	0.386	0.402	0.371	0.396	0.392	0.403	0.379	0.400	0.379	0.401
	192	0.465	0.449	0.444	0.431	0.433	0.429	0.421	0.424	0.441	0.431	0.427	0.428	0.429	0.432
	336	0.504	0.470	0.483	0.451	0.427	0.470	0.454	0.443	0.483	0.453	0.465	0.447	0.449	0.460
	720	0.502	0.492	0.486	0.477	0.470	0.472	0.474	0.462	0.484	0.477	0.460	0.462	0.468	0.460
	Avg	0.473	0.458	0.451	0.440	0.440	0.443	**0.430**	**0.431**	0.450	0.441	0.433	0.434	0.431	0.438
ETTh2	96	0.297	0.345	0.290	0.341	0.291	0.341	0.287	0.340	0.324	0.368	0.299	0.351	0.281	0.336
	192	0.388	0.400	0.371	0.390	0.369	0.389	0.389	0.391	0.405	0.415	0.377	0.400	0.362	0.387
	336	0.426	0.434	0.428	0.437	0.418	0.430	0.411	0.426	0.442	0.447	0.420	0.433	0.406	0.423
	720	0.431	0.446	0.430	0.446	0.423	0.443	0.421	0.442	0.463	0.467	0.422	0.446	0.418	0.441
	Avg	0.385	0.406	0.380	0.404	0.375	0.401	0.372	0.400	0.409	0.424	0.380	0.408	**0.367**	**0.397**
ETTm1	96	0.330	0.368	0.329	0.365	0.341	0.375	0.325	0.362	0.372	0.388	0.325	0.363	0.320	0.361
	192	0.369	0.385	0.370	0.385	0.382	0.393	0.366	0.383	0.408	0.406	0.368	0.383	0.361	0.383
	336	0.400	0.407	0.403	0.408	0.415	0.413	0.398	0.403	0.439	0.426	0.399	0.404	0.392	0.403
	720	0.460	0.439	0.459	0.438	0.470	0.444	0.459	0.436	0.496	0.458	0.462	0.439	0.453	0.438
	Avg	0.390	0.400	0.390	0.399	0.402	0.406	0.387	0.396	0.429	0.420	0.389	0.397	**0.382**	**0.396**
ETTm2	96	0.175	0.258	0.177	0.264	0.180	0.268	0.181	0.269	0.192	0.276	0.179	0.267	0.178	0.265
	192	0.247	0.307	0.241	0.304	0.245	0.309	0.244	0.308	0.255	0.314	0.244	0.307	0.243	0.305
	336	0.309	0.345	0.300	0.342	0.303	0.346	0.303	0.345	0.319	0.354	0.304	0.346	0.303	0.343
	720	0.408	0.403	0.398	0.398	0.399	0.401	0.402	0.400	0.416	0.406	0.402	0.401	0.400	0.399
	Avg	0.285	0.328	**0.279**	**0.327**	0.282	0.331	0.283	0.331	0.296	0.338	0.282	0.330	0.281	0.328
Weather	96	0.177	0.218	0.192	0.23	0.192	0.230	0.187	0.229	0.194	0.234	0.181	0.223	0.184	0.227
	192	0.225	0.259	0.237	0.267	0.237	0.267	0.231	0.265	0.237	0.268	0.227	0.261	0.230	0.264
	336	0.278	0.297	0.291	0.304	0.291	0.304	0.284	0.302	0.292	0.307	0.280	0.299	0.283	0.300
	720	0.354	0.348	0.365	0.353	0.365	0.353	0.358	0.348	0.362	0.351	0.354	0.346	0.356	0.348
	Avg	**0.259**	**0.281**	0.271	0.289	0.271	0.289	0.265	0.286	0.271	0.290	0.261	0.282	0.263	0.285
Exchange	96	0.084	0.201	0.191	0.309	0.178	0.300	0.176	0.297	0.190	0.308	0.175	0.297	0.177	0.300
	192	0.187	0.307	0.328	0.413	0.328	0.414	0.327	0.413	0.336	0.418	0.336	0.420	0.325	0.413
	336	0.337	9,422	0.853	0.696	0.864	0.700	0.850	0.696	0.945	0.741	0.842	0.692	0.842	0.691
	720	0.858	0.695	0.853	0.696	0.864	0.700	0.850	0.696	0.945	0.741	0.842	0.692	0.842	0.691
	Avg	0.367	0.406	0.365	0.406	0.364	0.404	0.359	**0.402**	0.391	0.420	0.359	0.403	**0.357**	0.403
ECL	96	0.193	0.291	0.239	0.329	0.218	0.305	0.183	0.273	0.211	0.297	0.180	0.270	0.177	0.262
	192	0.199	0.297	0.238	0.331	0.218	0.307	0.187	0.279	0.212	0.300	0.177	0.272	0.183	0.268
	336	0.216	0.312	0.254	0.345	0.232	0.321	0.203	0.294	0.227	0.314	0.201	0.284	0.198	0.283
	720	0.257	0.345	0.295	0.373	0.273	0.351	0.244	0.327	0.269	0.345	0.243	0.322	0.239	0.317
	Avg	0.216	0.311	0.257	0.345	0.235	0.321	0.204	0.293	0.230	0.314	0.200	0.287	**0.199**	**0.283**
Traffic	96	0.472	0.305	0.778	0.474	0.703	0.427	0.600	0.386	0.724	0.438	0.550	0.353	0.491	0.325
	192	0.474	0.304	0.730	0.455	0.651	0.408	0.573	0.375	0.677	0.420	0.563	0.365	0.486	0.317
	336	0.491	0.331	0.741	0.460	0.659	0.410	0.583	0.378	0.685	0.423	0.587	0.368	0.498	0.321
	720	0.523	0.327	0.780	0.475	0.699	0.427	0.619	0.395	0.724	0.439	0.602	0.385	0.532	0.338
	Avg	**0.490**	**0.317**	0.757	0.466	0.678	0.418	0.594	0.384	0.703	0.430	0.576	0.368	0.502	0.325

Table 18. In-domain fine-tuning results for time series classification. The model was pre-trained on datasets AD, TDBrain, and PTB, then fine-tuned on the same dataset. Accuracy (%), Precision (%), Recall (%), F1 score (%), AUROC (%), AUPRC (%) are recorded. We perform the experiment five times for each outcome and present the mean and standard deviation as our reported findings.

DATASETS	METHODS	ACCURACY	PRECISION	RECALL	F1 SCORE	AUROC	AUPRC
AD	RANDOM INIT.	80.62$_{\pm 2.17}$	80.51$_{\pm 2.24}$	80.48$_{\pm 2.18}$	80.48$_{\pm 2.19}$	86.60$_{\pm 1.60}$	86.48$_{\pm 1.74}$
	CPC	77.40$_{\pm 7.28}$	79.91$_{\pm 4.35}$	78.52$_{\pm 6.18}$	77.09$_{\pm 7.65}$	89.81$_{\pm 2.98}$	89.49$_{\pm 3.20}$
	TNC	78.58$_{\pm 6.21}$	81.10$_{\pm 4.09}$	79.97$_{\pm 5.50}$	78.43$_{\pm 6.35}$	92.26$_{\pm 2.38}$	92.10$_{\pm 2.60}$
	TS2VEC	81.26$_{\pm 2.08}$	81.21$_{\pm 2.14}$	81.34$_{\pm 2.04}$	81.12$_{\pm 2.06}$	89.20$_{\pm 1.76}$	88.94$_{\pm 1.85}$
	COST	73.87$_{\pm 4.35}$	77.22$_{\pm 2.36}$	75.51$_{\pm 3.70}$	73.60$_{\pm 4.65}$	89.28$_{\pm 2.07}$	88.78$_{\pm 2.23}$
	LAST	72.63$_{\pm 5.58}$	75.82$_{\pm 0.71}$	73.66$_{\pm 3.50}$	72.06$_{\pm 5.87}$	84.97$_{\pm 4.00}$	84.22$_{\pm 4.57}$
	TF-C	75.31$_{\pm 8.27}$	75.87$_{\pm 8.73}$	74.83$_{\pm 8.98}$	74.54$_{\pm 8.85}$	79.45$_{\pm 10.23}$	79.33$_{\pm 10.57}$
	COMET	84.50$_{\pm 4.46}$	88.31$_{\pm 2.42}$	82.95$_{\pm 5.39}$	83.33$_{\pm 5.15}$	94.44$_{\pm 2.37}$	94.43$_{\pm 2.48}$
	TST	81.50$_{\pm 2.16}$	82.23$_{\pm 2.12}$	82.35$_{\pm 2.16}$	81.49$_{\pm 2.16}$	90.41$_{\pm 2.06}$	89.67$_{\pm 2.42}$
	TI-MAE	80.70$_{\pm 3.73}$	82.23$_{\pm 2.92}$	81.84$_{\pm 3.39}$	80.67$_{\pm 3.73}$	92.32$_{\pm 2.80}$	92.18$_{\pm 2.93}$
	SIMMTM	86.19$_{\pm 1.12}$	87.08$_{\pm 1.42}$	85.41$_{\pm 1.03}$	85.89$_{\pm 1.11}$	91.99$_{\pm 0.83}$	92.04$_{\pm 0.84}$
	TIMESIAM	**89.93**$_{\pm 1.68}$	**90.23**$_{\pm 1.39}$	**89.46**$_{\pm 1.90}$	**89.72**$_{\pm 1.78}$	**95.31**$_{\pm 1.95}$	**95.25**$_{\pm 2.19}$
TDBRAIN	RANDOM INIT.	79.08$_{\pm 2.33}$	80.15$_{\pm 2.16}$	79.08$_{\pm 2.33}$	78.93$_{\pm 2.39}$	89.17$_{\pm 1.94}$	89.48$_{\pm 1.90}$
	CPC	85.19$_{\pm 2.99}$	85.35$_{\pm 2.88}$	85.19$_{\pm 2.99}$	85.17$_{\pm 3.01}$	93.50$_{\pm 2.55}$	93.68$_{\pm 2.50}$
	TNC	85.21$_{\pm 1.92}$	86.49$_{\pm 1.86}$	85.21$_{\pm 1.92}$	85.08$_{\pm 1.95}$	95.77$_{\pm 1.30}$	95.95$_{\pm 1.25}$
	TS2VEC	80.21$_{\pm 1.69}$	81.38$_{\pm 1.97}$	80.21$_{\pm 1.69}$	80.07$_{\pm 1.69}$	89.57$_{\pm 2.31}$	89.60$_{\pm 2.37}$
	COST	83.86$_{\pm 3.71}$	85.00$_{\pm 3.00}$	83.86$_{\pm 3.71}$	83.70$_{\pm 3.89}$	94.58$_{\pm 1.90}$	94.79$_{\pm 1.79}$
	LAST	85.13$_{\pm 1.85}$	85.79$_{\pm 1.54}$	85.13$_{\pm 1.85}$	85.06$_{\pm 1.90}$	94.88$_{\pm 8.26}$	95.10$_{\pm 0.81}$
	TF-C	66.62$_{\pm 1.76}$	67.15$_{\pm 1.64}$	66.62$_{\pm 1.76}$	66.35$_{\pm 1.91}$	65.43$_{\pm 6.13}$	66.18$_{\pm 4.90}$
	COMET	85.47$_{\pm 1.16}$	85.68$_{\pm 1.20}$	85.47$_{\pm 1.16}$	85.45$_{\pm 1.16}$	93.73$_{\pm 1.02}$	93.96$_{\pm 0.99}$
	TST	83.22$_{\pm 1.91}$	84.86$_{\pm 1.08}$	83.22$_{\pm 1.91}$	83.01$_{\pm 2.03}$	93.86$_{\pm 1.10}$	94.03$_{\pm 0.99}$
	TI-MAE	88.16$_{\pm 1.87}$	88.96$_{\pm 1.42}$	88.16$_{\pm 1.87}$	88.10$_{\pm 1.91}$	**97.27**$_{\pm 0.49}$	**96.94**$_{\pm 0.48}$
	SIMMTM	84.81$_{\pm 1.54}$	86.43$_{\pm 1.07}$	84.81$_{\pm 1.54}$	84.54$_{\pm 1.67}$	94.18$_{\pm 1.57}$	89.51$_{\pm 1.52}$
	TIMESIAM	**90.67**$_{\pm 1.24}$	**91.08**$_{\pm 1.13}$	**90.67**$_{\pm 1.24}$	**90.64**$_{\pm 1.25}$	96.96$_{\pm 0.80}$	96.82$_{\pm 0.82}$
PTB	RANDOM INIT.	84.19$_{\pm 1.29}$	83.35$_{\pm 1.68}$	78.46$_{\pm 2.50}$	80.33$_{\pm 2.02}$	89.55$_{\pm 1.83}$	83.61$_{\pm 2.68}$
	CPC	88.30$_{\pm 3.07}$	88.90$_{\pm 1.00}$	81.54$_{\pm 6.51}$	83.75$_{\pm 5.67}$	89.86$_{\pm 3.87}$	88.68$_{\pm 2.89}$
	TNC	90.53$_{\pm 2.92}$	89.01$_{\pm 2.87}$	87.06$_{\pm 5.22}$	87.82$_{\pm 4.13}$	93.12$_{\pm 2.21}$	91.01$_{\pm 1.55}$
	TS2VEC	85.14$_{\pm 1.66}$	87.82$_{\pm 2.21}$	76.84$_{\pm 3.99}$	79.66$_{\pm 3.63}$	90.50$_{\pm 1.59}$	90.07$_{\pm 1.73}$
	COST	88.61$_{\pm 1.36}$	87.75$_{\pm 1.23}$	80.23$_{\pm 2.39}$	83.81$_{\pm 2.33}$	93.79$_{\pm 2.36}$	93.01$_{\pm 2.37}$
	LAST	89.22$_{\pm 3.10}$	89.12$_{\pm 2.71}$	83.32$_{\pm 5.54}$	85.45$_{\pm 4.66}$	94.91$_{\pm 1.13}$	91.79$_{\pm 4.25}$
	TF-C	87.50$_{\pm 2.43}$	85.50$_{\pm 3.04}$	82.68$_{\pm 4.04}$	83.77$_{\pm 3.50}$	77.59$_{\pm 19.22}$	80.62$_{\pm 15.10}$
	COMET	87.84$_{\pm 1.98}$	87.67$_{\pm 1.72}$	81.14$_{\pm 3.68}$	83.45$_{\pm 3.22}$	92.95$_{\pm 1.56}$	87.47$_{\pm 2.82}$
	TST	84.25$_{\pm 3.29}$	84.05$_{\pm 3.95}$	74.83$_{\pm 5.49}$	77.45$_{\pm 5.59}$	90.44$_{\pm 3.05}$	85.74$_{\pm 3.25}$
	TI-MAE	88.39$_{\pm 1.78}$	88.55$_{\pm 1.51}$	81.76$_{\pm 3.13}$	84.23$_{\pm 2.70}$	90.37$_{\pm 5.70}$	88.76$_{\pm 5.00}$
	SIMMTM	90.04$_{\pm 1.23}$	89.09$_{\pm 1.33}$	85.52$_{\pm 2.57}$	87.05$_{\pm 2.38}$	92.68$_{\pm 1.23}$	90.14$_{\pm 3.01}$
	TIMESIAM	**91.32**$_{\pm 2.92}$	**89.97**$_{\pm 2.89}$	**88.02**$_{\pm 4.93}$	**88.84**$_{\pm 4.12}$	**96.42**$_{\pm 1.51}$	**94.33**$_{\pm 2.09}$

Table 19. Cross-domain fine-tuning results for time series classification. The model is pre-trained on the TSLD-1G dataset and fine-tuned on AD, TDBrain, and PTB. Accuracy (%), Precision (%), Recall (%), F1 score (%), AUROC (%), AUPRC (%) are recorded. We perform the experiment five times for each outcome and present the mean and standard deviation as our reported findings.

DATASETS	METHODS	ACCURACY	PRECISION	RECALL	F1 SCORE	AUROC	AUPRC
AD	RANDOM INIT.	80.62$_{\pm 2.17}$	80.51$_{\pm 2.24}$	80.48$_{\pm 2.18}$	80.48$_{\pm 2.19}$	86.60$_{\pm 1.60}$	86.48$_{\pm 1.74}$
	TS2VEC	80.59$_{\pm 6.45}$	81.77$_{\pm 8.72}$	81.61$_{\pm 9.20}$	80.53$_{\pm 9.55}$	90.31$_{\pm 6.38}$	90.00$_{\pm 7.94}$
	TF-C	87.98$_{\pm 1.77}$	88.30$_{\pm 1.68}$	88.30$_{\pm 1.69}$	87.90$_{\pm 1.75}$	95.56$_{\pm 1.52}$	95.43$_{\pm 1.54}$
	TST	82.60$_{\pm 3.71}$	83.81$_{\pm 2.63}$	83.35$_{\pm 3.02}$	82.51$_{\pm 3.66}$	93.05$_{\pm 2.17}$	92.75$_{\pm 2.50}$
	TI-MAE	80.40$_{\pm 5.26}$	81.72$_{\pm 5.02}$	81.13$_{\pm 4.66}$	80.31$_{\pm 5.18}$	91.32$_{\pm 4.48}$	91.16$_{\pm 4.75}$
	SIMMTM	87.74$_{\pm 1.78}$	87.66$_{\pm 1.91}$	87.78$_{\pm 1.78}$	87.63$_{\pm 1.78}$	94.73$_{\pm 1.32}$	94.71$_{\pm 1.36}$
	TIMESIAM	**90.47**$_{\pm 2.04}$	**90.50**$_{\pm 2.01}$	**90.21**$_{\pm 2.13}$	**90.32**$_{\pm 2.09}$	**96.34**$_{\pm 1.36}$	**96.41**$_{\pm 1.39}$
TDBRAIN	RANDOM INIT.	79.08$_{\pm 2.33}$	80.15$_{\pm 2.16}$	79.08$_{\pm 2.33}$	78.93$_{\pm 2.39}$	89.17$_{\pm 1.94}$	89.48$_{\pm 1.90}$
	TS2VEC	85.58$_{\pm 8.16}$	86.26$_{\pm 7.78}$	85.58$_{\pm 8.16}$	85.45$_{\pm 8.32}$	94.44$_{\pm 4.03}$	94.69$_{\pm 3.79}$
	TF-C	82.84$_{\pm 2.57}$	83.22$_{\pm 2.58}$	82.84$_{\pm 2.57}$	82.79$_{\pm 2.58}$	92.13$_{\pm 2.17}$	92.28$_{\pm 2.11}$
	TST	83.65$_{\pm 2.52}$	84.75$_{\pm 2.27}$	83.65$_{\pm 2.52}$	83.51$_{\pm 2.59}$	93.41$_{\pm 2.13}$	93.58$_{\pm 2.08}$
	TI-MAE	85.22$_{\pm 2.40}$	82.85$_{\pm 2.01}$	82.42$_{\pm 2.47}$	82.38$_{\pm 2.53}$	90.25$_{\pm 1.39}$	90.26$_{\pm 1.36}$
	SIMMTM	85.29$_{\pm 1.87}$	86.61$_{\pm 2.45}$	**86.23**$_{\pm 2.59}$	**86.19**$_{\pm 2.62}$	94.00$_{\pm 1.78}$	93.95$_{\pm 1.84}$
	TIMESIAM	**86.26**$_{\pm 2.54}$	**87.17**$_{\pm 2.07}$	80.26$_{\pm 2.54}$	86.17$_{\pm 2.62}$	**95.41**$_{\pm 1.35}$	**95.56**$_{\pm 1.30}$
PTB	RANDOM INIT.	84.19$_{\pm 1.29}$	83.35$_{\pm 1.68}$	78.46$_{\pm 2.50}$	80.33$_{\pm 2.02}$	89.55$_{\pm 1.83}$	83.61$_{\pm 2.68}$
	TS2VEC	89.23$_{\pm 7.76}$	89.58$_{\pm 7.15}$	**89.23**$_{\pm 7.76}$	**89.17**$_{\pm 7.88}$	**96.13**$_{\pm 3.82}$	**96.27**$_{\pm 3.63}$
	TF-C	89.18$_{\pm 1.89}$	88.63$_{\pm 2.21}$	84.48$_{\pm 3.98}$	85.64$_{\pm 2.66}$	94.31$_{\pm 1.71}$	91.52$_{\pm 2.79}$
	TST	85.81$_{\pm 5.92}$	85.80$_{\pm 4.96}$	77.40$_{\pm 1.02}$	79.32$_{\pm 1.07}$	92.04$_{\pm 2.89}$	86.27$_{\pm 4.97}$
	TI-MAE	86.67$_{\pm 2.55}$	85.91$_{\pm 1.10}$	80.32$_{\pm 6.76}$	81.83$_{\pm 4.94}$	92.60$_{\pm 4.45}$	91.08$_{\pm 3.78}$
	SIMMTM	85.64$_{\pm 1.68}$	85.94$_{\pm 1.58}$	77.01$_{\pm 3.00}$	79.80$_{\pm 2.78}$	92.93$_{\pm 0.68}$	88.03$_{\pm 2.31}$
	TIMESIAM	**90.45**$_{\pm 1.98}$	**89.58**$_{\pm 1.53}$	86.11$_{\pm 3.75}$	87.53$_{\pm 2.83}$	93.13$_{\pm 2.32}$	89.94$_{\pm 3.10}$

Table 20. Ablation studies were conducted on TimeSiam. "W/o Siamese" refers to solely focusing on modeling subseries itself, without incorporating Siamese modeling. "W/o Masking" indicates the absence of mask augmentation in the current subseries.

INPUT-96		RANDOM INIT.		W/O SIAMESE		W/O MASKING		**TIMESIAM**	
PREDICT-*O*		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTH1	96	0.420	0.423	0.377	0.401	0.381	0.403	0.378	0.401
	192	0.465	0.449	0.423	0.430	0.430	0.431	0.422	0.430
	336	0.504	0.470	0.458	0.451	0.466	0.452	0.459	0.452
	720	0.502	0.492	0.471	0.478	0.470	0.474	0.459	0.437
	AVG	0.473	0.458	0.432	0.440	0.437	0.440	**0.429**	**0.437**
ETTH2	96	0.297	0.345	0.291	0.346	0.289	0.339	0.293	0.345
	192	0.388	0.400	0.375	0.396	0.378	0.393	0.370	0.392
	336	0.426	0.434	0.416	0.432	0.412	0.426	0.410	0.424
	720	0.431	0.446	0.421	0.446	0.420	0.441	0.418	0.440
	AVG	0.385	0.406	0.376	0.405	0.375	0.400	**0.373**	**0.400**
ETTM1	96	0.330	0.368	0.317	0.359	0.333	0.368	0.319	0.360
	192	0.369	0.385	0.363	0.387	0.367	0.385	0.353	0.379
	336	0.400	0.407	0.385	0.403	0.400	0.409	0.383	0.402
	720	0.460	0.439	0.444	0.438	0.459	0.442	0.440	0.436
	AVG	0.390	0.400	0.377	0.397	0.390	0.410	**0.374**	**0.394**
ETTM2	96	0.175	0.258	0.177	0.262	0.177	0.261	0.175	0.261
	192	0.247	0.307	0.241	0.303	0.243	0.303	0.241	0.303
	336	0.309	0.345	0.302	0.343	0.307	0.347	0.300	0.341
	720	0.408	0.403	0.398	0.398	0.405	0.404	0.399	0.398
	AVG	0.285	0.328	0.280	0.327	0.283	0.329	**0.279**	**0.326**
WEATHER	96	0.177	0.218	0.174	0.219	0.176	0.219	0.171	0.213
	192	0.225	0.259	0.221	0.258	0.224	0.259	0.217	0.253
	336	0.278	0.297	0.275	0.296	0.279	0.299	0.272	0.293
	720	0.354	0.384	0.353	0.345	0.356	0.350	0.348	0.343
	AVG	0.259	0.281	0.256	0.280	0.259	0.282	**0.252**	**0.276**
EXCHANGE	96	0.084	0.201	0.089	0.209	0.083	0.201	0.084	0.203
	192	0.187	0.307	0.196	0.314	0.173	0.297	0.176	0.300
	336	0.337	0.422	0.334	0.419	0.341	0.424	0.310	0.404
	720	0.858	0.695	0.856	0.700	0.856	0.698	0.842	0.690
	AVG	0.367	0.406	0.369	0.411	0.363	0.405	**0.353**	**0.399**
ECL	96	0.193	0.291	0.164	0.250	0.165	0.253	0.164	0.245
	192	0.199	0.297	0.175	0.261	0.177	0.258	0.173	0.256
	336	0.216	0.312	0.191	0.278	0.190	0.274	0.189	0.275
	720	0.257	0.345	0.230	0.312	0.232	0.311	0.229	0.310
	AVG	0.216	0.331	0.190	0.275	0.191	0.274	**0.189**	**0.272**
TRAFFIC	96	0.472	0.305	0.433	0.281	0.438	0.283	0.429	0.279
	192	0.474	0.304	0.446	0.287	0.447	0.287	0.442	0.282
	336	0.491	0.331	0.459	0.288	0.459	0.289	0.456	0.288
	720	0.523	0.327	0.490	0.306	0.494	0.307	0.486	0.307
	AVG	0.490	0.317	0.457	0.291	0.460	0.292	**0.453**	**0.289**